

AD-A227 651	DTIC ACCESSION NUMBER		PHOTOGRAPH THIS SHEET		1																																																																														
		LEVEL	DTIC FILE COPY		INVENTORY																																																																														
		<div style="font-size: 1.5em; font-weight: bold;">WRDC-TR-90-8022</div> <div style="font-size: 0.8em;">DOCUMENT IDENTIFICATION</div> <div style="font-size: 1.2em; font-weight: bold;">NOV 1990</div>																																																																																	
<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: 80%;"> <b>DISTRIBUTION STATEMENT A</b>              Approved for public release              Distribution Unlimited           </div>																																																																																			
DISTRIBUTION STATEMENT																																																																																			
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="6" style="text-align: left; font-size: 0.7em;">ACCESSION FOR</td> </tr> <tr> <td style="width: 15%;">NTIS</td> <td style="width: 15%;">GRA&amp;I</td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> </tr> <tr> <td>DTIC</td> <td>TRAC</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td colspan="6">UNANNOUNCED</td> </tr> <tr> <td colspan="6">JUSTIFICATION</td> </tr> <tr> <td colspan="6" style="height: 20px;"></td> </tr> <tr> <td colspan="6" style="height: 20px;"></td> </tr> <tr> <td colspan="6" style="height: 20px;"></td> </tr> <tr> <td colspan="6">BY</td> </tr> <tr> <td colspan="6">DISTRIBUTION/</td> </tr> <tr> <td colspan="6">AVAILABILITY CODES</td> </tr> <tr> <td colspan="2">DISTRIBUTION</td> <td colspan="4">AVAILABILITY AND/OR SPECIAL</td> </tr> <tr> <td colspan="2" style="height: 40px; vertical-align: bottom; font-size: 1.5em;">A-1</td> <td colspan="4"></td> </tr> </table>						ACCESSION FOR						NTIS	GRA&I					DTIC	TRAC					UNANNOUNCED						JUSTIFICATION																								BY						DISTRIBUTION/						AVAILABILITY CODES						DISTRIBUTION		AVAILABILITY AND/OR SPECIAL				A-1					
ACCESSION FOR																																																																																			
NTIS	GRA&I																																																																																		
DTIC	TRAC																																																																																		
UNANNOUNCED																																																																																			
JUSTIFICATION																																																																																			
BY																																																																																			
DISTRIBUTION/																																																																																			
AVAILABILITY CODES																																																																																			
DISTRIBUTION		AVAILABILITY AND/OR SPECIAL																																																																																	
A-1																																																																																			
<div style="border: 1px solid black; padding: 10px; width: 150px; margin: 0 auto;"> <div style="font-size: 2em; font-weight: bold; text-align: center;">S</div> <div style="font-size: 1.5em; font-weight: bold; text-align: center;">DTIC ELECTE</div> <div style="font-size: 1.2em; text-align: center;">OCT 22 1990</div> <div style="font-size: 2em; font-weight: bold; text-align: center;">D</div> <div style="font-size: 1.5em; font-weight: bold; text-align: center;">E</div> </div>																																																																																			
DATE ACCESSIONED																																																																																			
DATE RETURNED																																																																																			
REGISTERED OR CERTIFIED NUMBER																																																																																			
<div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 400px;"> <div style="font-size: 1.5em; font-weight: bold; text-align: center;">90 09 13 005</div> </div>																																																																																			
DATE RECEIVED IN DTIC																																																																																			
PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC																																																																																			

GEOMETRIC MODELING APPLICATIONS INTERFACE PROGRAM

TECHNICAL TRANSFER DOCUMENT PRODUCT INFORMATION  
EXCHANGE SYSTEM USER'S MANUAL



United Technologies Corporation  
Pratt and Whitney  
Government Products Division  
P.O. Box 9600  
West Palm Beach Florida 33410-9600

November 1990

Final Report for Period August 1985 - March 1989

Approved for public release; distribution is unlimited

MANUFACTURING TECHNOLOGY DIRECTORATE  
WRIGHT RESEARCH AND DEVELOPMENT CENTER  
AIR FORCE SYSTEMS COMMAND  
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

## REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) FR 20674			5. MONITORING ORGANIZATION REPORT NUMBER(S) WRDC-TR-90-8022		
6a. NAME OF PERFORMING ORGANIZATION United Technologies Corporation Pratt & Whitney Government Products Division		6b. OFFICE SYMBOL (If applicable) (P&W)	7a. NAME OF MONITORING ORGANIZATION Wright Research and Development Center Manufacturing Technology Directorate (cont)		
6c. ADDRESS (City, State and ZIP Code) P.O. Box 9600 West Palm Beach, Florida 33412-9600		7b. ADDRESS (City, State and ZIP Code) Wright-Patterson AFB OH 45433-6533			
8a. NAME OF FUNDING/SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33615-85-C-5122		
8c. ADDRESS (City, State and ZIP Code)		10. SOURCE OF FUNDING NOS.			
		PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
		78011F	MTPI	06	84
11. TITLE (Include Security Classification) Geometric Modeling Applications Interface (cont)					
12. PERSONAL AUTHOR(S) R. Helldoerfer, C. Van Wie, D. Emmerson					
13a. TYPE OF REPORT Final		13b. TIME COVERED 31 Mar FROM 1 Aug 85 TO 89		14. DATE OF REPORT (Yr., Mo., Day) November 1990	
15. PAGE COUNT 79					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB. GR.	Geometric Modeling Applications Interface Program		
			Product Definition Data Interface		
			Turbine Blades and Disks		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)  This User Manual provides a description of the data exchange software developed under the Air Force sponsored Product Definition Data Interface contract (F33615-82-C-5036) and is applicable to the GMAP contract (F33615-85-C-5122), which is sponsored by the Computer Integrated Manufacturing Branch, Materials Laboratory, Air Force Systems Command, Wright Air Force Base, Ohio.					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>			21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL David Judson		22b. TELEPHONE NUMBER (Include Area Code) (513) 255-7371		22c. OFFICE SYMBOL WRDC/MTI	

7a. (WRDC/MTI)

11. Program


18. Product Life Cycle  
Engineering  
Manufacturing  
Interface  
Exchange Format  
CAD  
CAM  
CIM  
IBIS  
RFC  
System Translator  
Schema Manager  
Model Access Software  
Name/Value Interface

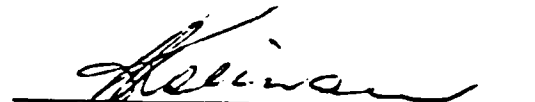
## NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

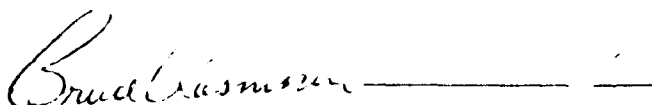
This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

  
Charles Gilman  
Project Manager

  
Walter H. Reimann, Chief  
Computer-Integrated Mfg. Branch

FOR THE COMMANDER

  
BRUCE A. RASMUSSEN  
Chief, Integration Technology Division  
Manufacturing Technology Directorate

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, WPAFB, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

## FOREWORD

This User Manual describes work performed under Air Force Contract F33615-85-C-5122, Geometric Modeling Applications Interface Program (GMAP), covering the period 1 August 1985 to 31 March 1989. This User Manual provides a description of the data exchange software developed under the Air Force sponsored Product Definition Data Interface contract (F33615-82-C-5036). This exchange software is applicable to the GMAP contract which is sponsored by the Computer Integrated Manufacturing Branch, Materials Laboratory, Air Force Systems Command, Wright Air Force Base, Ohio 45433-6533. The GMAP Project Manager for the Air Force is Mr. Charles Gilman.

The primary contractor is Pratt & Whitney, an operating unit of United Technologies Corporation. Mr. Richard Lopatka is managing the GMAP project at Pratt & Whitney. Ms. Linda Phillips is the Program Integrator. Mr. John Hamill is the Deputy Program Manager.

McDonnell Aircraft Company is the subcontractor responsible for developing the Product Information Exchange System software. Mr. Jerry Weiss is the GMAP Program Manager at McDonnell Aircraft and Mr. Herb Ryan is the Deputy Program Manager.

NOTE: The number and date in the upper right corner of each page in this document indicate that it has been prepared in accordance to the ICAM CM Life Cycle Documentation requirements for a Configuration Item (CI).

July 1989

TABLE OF CONTENTS

		<u>Page</u>
SECTION 1.	SCOPE.....	1-1
1.1	Identification.....	1-1
1.2	Functional Summary.....	1-1
1.3	Approach.....	1-2
SECTION 2.	PRODUCT INFORMATION EXCHANGE	
	SYSTEM (PIES) FUNCTIONS.....	2-1
2.1	Overview.....	2-1
2.2	File Allocations.....	2-1
2.3	User Functions.....	2-3
2.3.1	Overview.....	2-3
2.3.2	PIES Main Menu.....	2-4
2.3.3	System Translator.....	2-4
2.3.3.1	Overview.....	2-4
2.3.3.2	System Translator Menu.....	2-4
2.3.3.3	Exchange Format File Description.....	2-5
2.3.4	PDDI Interim Database.....	2-7
2.3.4.1	Overview.....	2-7
2.3.4.2	PDDI Interim Database Functions.....	2-8
2.3.4.2.1	Access PDDI Interim Database.....	2-8
2.3.4.2.2	File the Working Form Model.....	2-8
2.3.4.2.3	Retrieve the Working Form Model.....	2-8
2.3.4.3	The PDDI Interim Database Software as a Separate Subsystem.....	2-8
2.3.5	Query.....	2-9
2.3.5.1	Overview.....	2-9
2.3.5.2	Enter "HEXON" to Print Application Data Block ADB in Hexadecimal Form.....	2-11
2.3.5.3	Enter "MODSPA" to Produce a Model Space Listing.....	2-11
2.3.5.4	Enter "ALL" to Query All Data.....	2-14
2.3.5.5	Enter "KIND" to Query Data by Entity kind.....	2-14
2.3.5.6	Enter "NETREF" to Query Inclusive	
	Entity References.....	2-15
2.3.5.7	Enter "NETDAT" to Query Inclusive Entity Data.....	2-19
2.3.5.8	Enter "ID" to Query Data by Entity.....	2-20
2.3.6	Verify.....	2-20
2.3.6.1	Overview.....	2-20

TABLE OF CONTENTS (contd.)

		<u>Page</u>
SECTION 3.	BATCH PROCESSING.....	3-1
3.1	IBM Batch Processing of PIES.....	3-1
3.2	Virtual Extended Architecture (VAX) Batch Processing of PIES.....	3-5
SECTION 4.	PIES OPERATOR's MANUAL.....	4-1
4.1	Introduction.....	4-1
4.2	Operations.....	4-8
4.2.1	Introduction.....	4-1
4.2.2	Files.....	4-1
4.2.3	Operator Interface.....	4-1
4.2.4	Compile.....	4-1
4.2.5	Link.....	4-1
4.3	Installation Procedures.....	4-2
4.3.1	Introduction.....	4-2
4.3.2	Preliminary Steps.....	4-2
4.3.3	Procedures - PIES Software.....	4-2
SECTION 5.	PERFORMANCE.....	5-1
APPENDIX A	REFERENCES.....	A-1



LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1-1	Product Information Exchange System.....	1-2
1-2	Product Definition Data Interface Translator.....	1-3
1-3	Direct Exchange Between Systems.....	1-6
1-4	Neutral Exchange Between Systems.....	1-7
1-5	Translation Method.....	1-9
5-1	Translator and PID Performance on the Turned Part Model.....	5-2
5-2	Translator and PID Performance on the Machine Rib Model.....	5-3

LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
4-1	PIES Software Tape JCL IBM/MVS.....	4-5
4-2	PIES Software Tape DCL VAX/VM.....	4-6
4-3	PIES Software Components.....	4-7
4-4	PIES IBM JCL and CLISTS.....	4-8
4-5	PIES VAX Command File.....	4-8

## SECTION 1

### SYSTEM OVERVIEW

#### 1.1 Introduction

The Product Information Exchange System is a computer independent application that uses the data exchange software developed for the PDDI and GMAP programs. It showcases the Model Access Software, Product Definition Data Interface Interim Database (PID), and System Translator subsystems of GMAP. It allows a user to access/store/query/verify data from a Product Data Exchange Specification Structured Exchange Format file or a Product Definition Data Interface Interim Database file, as shown in Figure 1-1.

The Product Information Exchange System was developed by McDonnell Aircraft Company for the Product Definition Data Interface. Pratt & Whitney personnel utilize an equivalent system for GMAP. It includes identical software with additional components, such as the GMAP Product Definition Data Editor.

#### 1.2 Overview

The Product Information Exchange System software, as well as the GMAP subsystem software, is designed to be computer and data independent. The use of high level languages, such as PASCAL and FORTRAN, facilitated software transportability. The use of a Data Dictionary unique to GMAP/Product Definition Data Interface allowed data independence.

The key ingredient of the Product Information Exchange System software is the Model Access Software. Model Access Software is an integrated set of routines that create and manage a memory resident Working Form (NETWORK) Model through key access. Model Access Software enables the application to be independent of the actual physical storage of the Working Form data. For a more detailed discussion of Model Access Software, refer to the Model Access Software User's Manual (CI UM560240031U) and the GMAP System Components As-Built Product Specification (CI PS560240032U). A complete listing of Air Force approved documents is presented in Appendix A.

The main option available through the Product Information Exchange System software is the GMAP/Product Definition Data Interface Translator. The Product Definition Data Interface Translator is the software package used to transmit the product definition data between systems using a PDES structured file. The preprocessor portion of the translator provides the interface from the Model Access Software Working Form to the Product Data Exchange Specification Exchange Format file. The post-processor portion of the translator provides the interface from the PDES Exchange Format file to the Model Access Software Working Form, as illustrated in Figure 1-2. A PID

file refers to a Working Form that has been temporarily filed to a disk or work area. For a more detailed discussion of the Product Definition Data Interface Translator, refer to the System Translator 'User's Manual (CI UM560240021U) and the GMAP System Components As-Built Product Specification (CI PS560240032U).

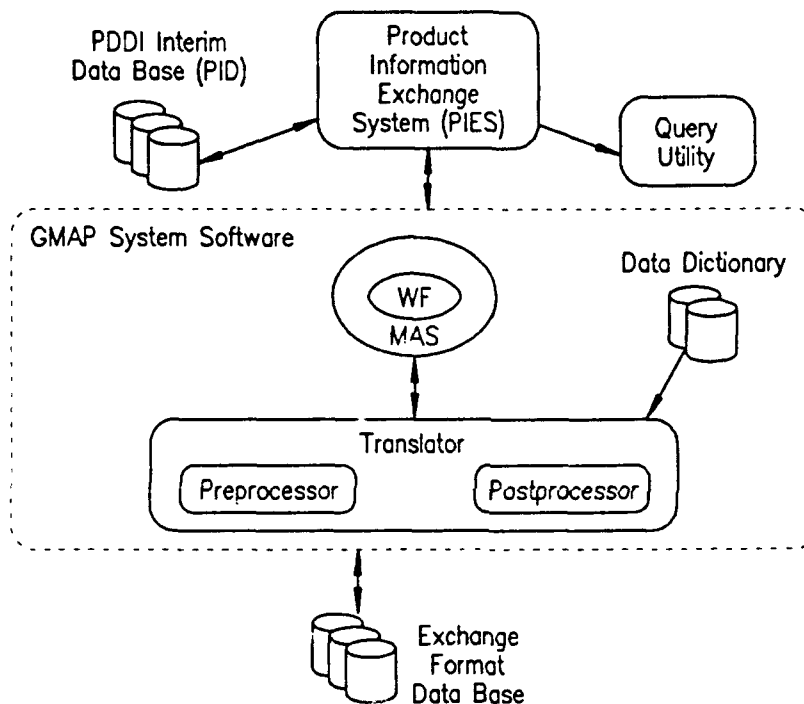


Figure 1-1. Product Information Exchange System

July 1989

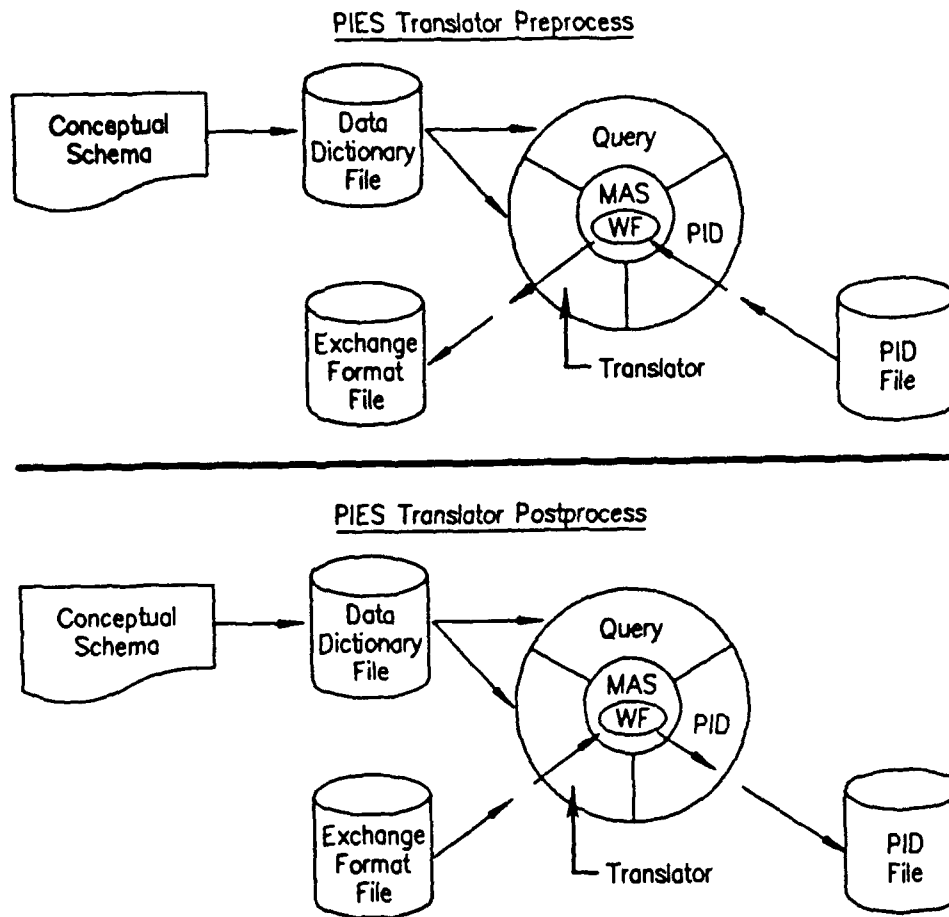


Figure 1-2. Product Definition Data Interface Translator

July 1989

Another option available through the Product Information Exchange System is the ability to store and access Working Form models from disk using the PID. The PID software will file a Working Form model to a sequential file or retrieve that file and recreate the Working Form model. PID can be used to perform a direct transfer of product definition data between similar computers (IBM-IBM, VAX-VAX, SUN-SUN, and so on). This software was originally designed as an intermediate storage facility for the Working Form while development continued on the translator(s) to the receiving system's native data base(s).

Two other Product Information Exchange System options are the ability to query data in the model or verify that the model was created as intended.

### 1.3 Data Exchange Methods

#### 1.3.1 Overview

In data exchange, there are several means to an end. Two typical methods of data exchange are presented in this section. Each is given a brief explanation and evaluated in terms of its advantages and disadvantages.

#### 1.3.2 Direct Exchange

The Direct Exchange method of translation is illustrated in Figure 1-3. As illustrated, if an exchange between  $n$  systems is required, then  $n \cdot (n-1)$  translators need to be developed to provide for the exchange.

##### Advantages:

- o In most cases, this method proves to be very fast.
- o It is potentially more accurate than any other type of exchange.

##### Disadvantages:

- o It requires a detailed knowledge of all systems involved in the exchange.
- o Each translator is very dependent on the others, making them sensitive to change. If a new system enters the exchange environment, all systems must develop additional translators to exchange with the new system.

- o The need to create and maintain a variety of direct translators places a considerable burden on a company's personnel and technical resources. The amount of detail required to understand a translator limits an individual to only understanding (i.e., supporting) a limited number.
- o Several complexities exist when attempting to exchange data between systems on dissimilar computers.

### 1.3.3 Neutral Exchange

Figure 1-4 illustrates the Neutral Exchange method of data exchange. This method involves developing translators that interface only between one system and a neutral exchange file. In contrast to Direct Exchange, if  $n$  systems require to exchange data,  $2n$  translators are required in the exchange. The form of the Neutral Exchange file usually is one of the exchange standards that have been developed by industry (i.e., EDIF, Initial Graphics Exchange Specification, IPC, Product Data Exchange Specification, VHDL, and so on).

#### Advantages:

- o Each translator is not dependent on the environment or data structures of any of the other translators. A new translator entering the exchange environment does not affect any other system.
- o Because there are fewer translators involved, maintenance is considerably less than by Direct Exchange.
- o The exchange file is character-based and is therefore portable between dissimilar computer environments (i.e., IBM/VAX, IBM/CDC, VAX/CDC, and so on).

#### Disadvantages:

- o The application developer needs to know the physical structure of the Exchange Format file.
- o It is not as fast as the Direct Exchange method.
- o It potentially has less accurate data conversion than the Direct Exchange method.
- o The opportunity to share common functional software is slightly greater than the Direct Exchange method.

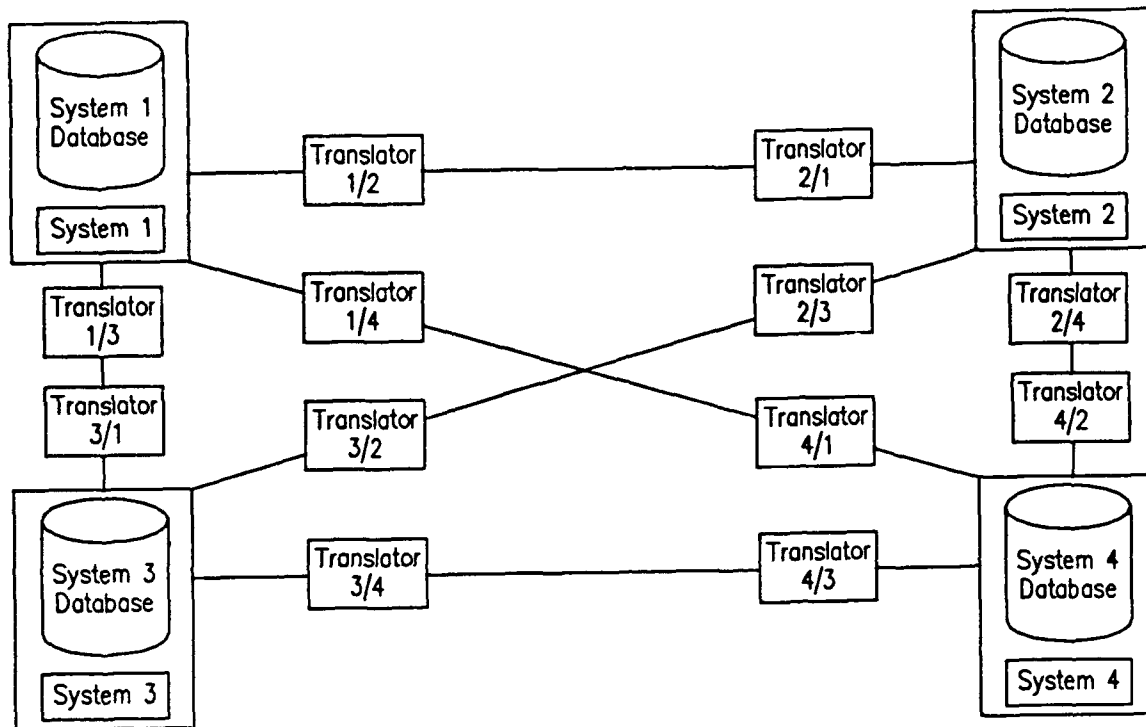


Figure 1-3. Direct Exchange Between Systems



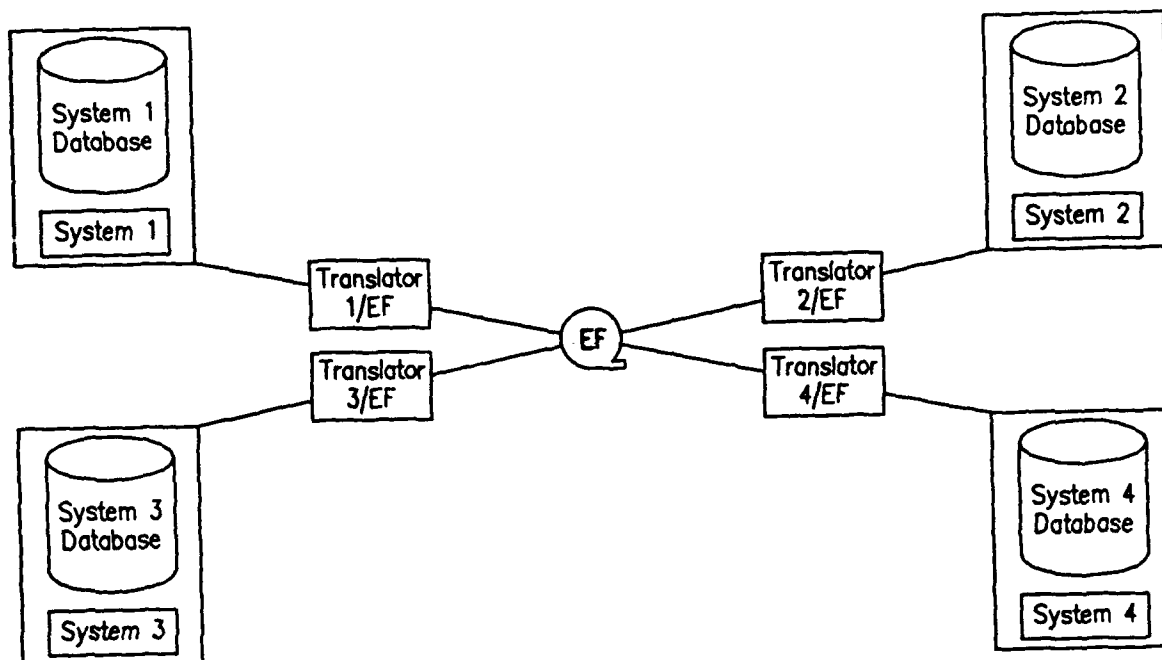


Figure 1-4. Neutral Exchange Between Systems

#### 1.3.4 Variations of Neutral Exchange

Because of the large amount of work and maintenance involved with maintaining a direct translator, the Neutral Exchange was chosen as the method of translation for the Product Definition Data Interface and GMAP. But there are many variations of Neutral Exchange. The following is a discussion of major variations and a discussion of the reasoning behind the Product Definition Data Interface method.

##### 1.3.4.1 Direct Neutral Exchange

The Direct Neutral Exchange method of translation is represented in the upper right portion of Figure 1-5, "Translation Methods." A Direct Neutral Exchange is a method in which each user writes a translator to interface between their computer system and a neutral Exchange Format file. This file is then used by others for the exchange of data. In today's environment, the most common form of this Exchange Format file is the Initial Graphic Exchange Specification. In the future, the form of this Exchange Format file will be the Product Data Exchange Specification.

##### Advantages:

- o Because there are fewer translators involved, maintenance is considerably less than the Direct Exchange method.

##### Disadvantages:

- o The application developer needs to know the physical structure of the Exchange Format file.
- o It is not as fast as the Direct Exchange method.
- o Potentially less accurate data conversion than the Direct Exchange method.
- o The opportunity to share common functional software is slightly greater than the Direct Exchange method.

##### 1.3.4.2 Database Neutral Exchange

The Database Neutral Exchange method of translation is represented in the bottom left portion of Figure 1-5. A Database Neutral Exchange is a method in which each user writes a translator to interface between their computer system and an agreed upon database that contains all of the kinds of information that is required to be transferred.

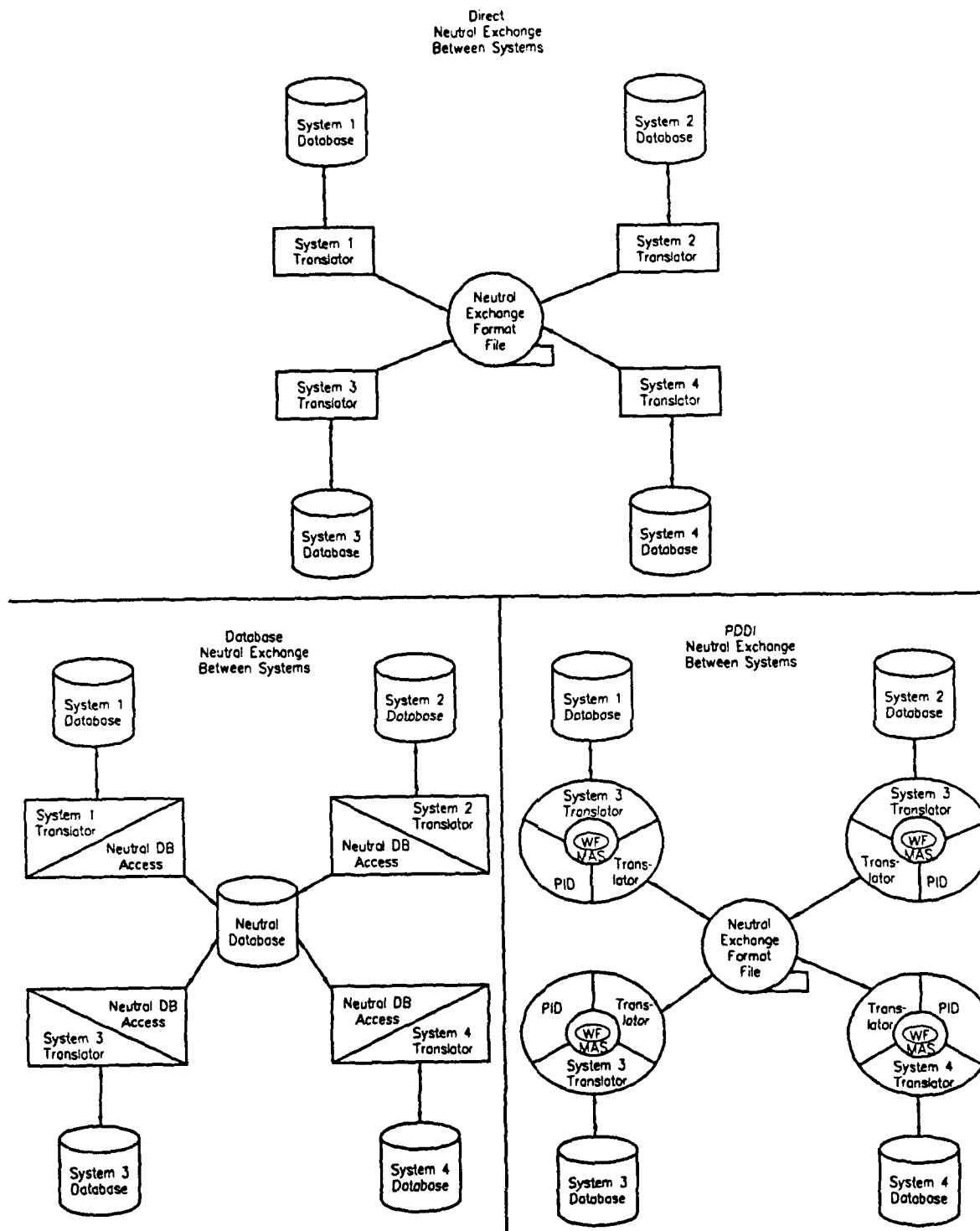


Figure 1-5. Translation Method

Advantages:

- o It provides a standard database capabilities for the data; the greatest opportunity to share common functional software and reduce translator development time.
- o Because there are fewer translators involved, maintenance is considerably less than the Direct Exchange method.

Disadvantages:

- o Additional complexities exist when accessing data from dissimilar computers (particularly in the areas of version control, data integrity, and data security).
- o It makes it very difficult to get an agreement on one database.

1.3.4.3 Product Definition Data Interface Exchange

The Product Definition Data Interface method of translation is represented in the Bottom right portion of Figure 1-5. In this method, each user writes a translator to interface between their computer system and an In-Memory (Working Form) neutral representation of the data to be transferred. Standard software is provided that will allow the user access to this Working Form model of the transfer data.

Advantages:

- o If one system involved in the transfer changes, only the translator that operates from that system needs to change.
- o The user does not need to know the physical structure of the Exchange Format file used to transfer the data.
- o It is extensible to any type of data by simply modifying the external file.
- o The Exchange Format file is character based and transportable between dissimilar computers.
- o It provides standard access capabilities to the data, providing a great opportunity to share common functional software and reduce translator development time.

Disadvantages:

- o It is not as fast as the Direct Exchange method.

- o It potentially offers less accurate data conversion than the Direct Exchange method.

#### 1.3.5 Summary

By comparing the different methods of data exchange, one can clearly see that the Product Definition Data Interface Exchange approach is superior. The Product Definition Data Interface Exchange method has more advantages and fewer liabilities than any other approach that is feasible using current technology.

Naturally, each user must weigh the advantages and disadvantages of each method and choose the method that best suits individual requirements. The current industry trend is to try to progress toward the Database Neutral Exchange method.

While the appeal of the Database Neutral Exchange method is considerable, technology is not yet available. The Product Definition Data Interface Exchange method is seen as a necessary step in the evolution toward that technology, while still meeting the production needs of today's industry.

July 1989

## SECTION 2

### PRODUCT INFORMATION EXCHANGE SYSTEM (PIES) FUNCTIONS

#### 2.1 Overview

PIES is a computer independent application that demonstrates the data exchange software developed for the Product Definition Data Interface (PDDI) and GMAP programs. It showcases the Model Access Software (MAS), PDDI Interim Database (PID), and System Translator subsystems of GMAP. It allows a user to access/store/query/verify data from a Product Data Exchange Specification (PDES) Structured Exchange Format file or a PID file (refer to Figure 1-1). The PIES software provides a standard means of transferring and accessing the data to be exchanged from an alphanumeric terminal.

#### 2.2 File Allocations

Since the PIES software needed to be computer transportable, no system dependent terminal dialog manager was used. This software uses PASCAL and FORTRAN reads and writes to interact with the user. Any file allocations that are needed are performed in the CLIST(IBM), JCL(IBM-BATCH), or DCL(VAX) prior to entry into the program.

The allocations that were used in the operation of the software are as follows:

1. FILE: DDFILE  
ALLOCATIONS: RECORD FORMAT = FB(FIXED BLOCK)  
RECORD LENGTH = 80 BYTES  
BLOCK SIZE = 80 BYTES  
FILE NAME: CAD5.GMAP.V40.DDDEFN.DATA (or user supplied)  
FUNCTION: DATA DICTIONARY INPUT
2. FILE: DDINX  
ALLOCATIONS: RECORD FORMAT = FB(FIXED BLOCK)  
RECORD LENGTH = 80 BYTES  
BLOCK SIZE = 6160 BYTES  
FILE NAME: CAD5.GMAP.V40.DDINDEX.DATA (or user supplied)  
FUNCTION: INDEX FILE FOR RELATIVE ACCESS TO THE  
DATA DICTIONARY FILE
3. FILE: EFFILE  
ALLOCATIONS: RECORD FORMAT = FB(FIXED BLOCK)  
RECORD LENGTH = 80 BYTES  
BLOCK SIZE = 6160 BYTES  
FILE NAME: USER SUPPLIED PRIOR TO EXECUTION

FUNCTION: EXCHANGE FORMAT FILE  
FOR INPUT OR OUTPUT

4. FILE: FT08F001  
ALLOCATIONS: RECORD FORMAT = FB(FIXED BLOCK)  
RECORD LENGTH = 80 BYTES  
BLOCK SIZE = 6160 BYTES  
FILE NAME: USER INPUT  
FUNCTION: PDDI INTERIM DATABASE (PID) FILE  
FOR INPUT OR OUTPUT

5. FILE: INPUT  
ALLOCATIONS: RECORD FORMAT = NA  
RECORD LENGTH = 80 BYTES  
BLOCK SIZE = NA  
FILE NAME: TERMINAL SCREEN  
FUNCTION: INPUT FROM TERMINAL SCREEN

6. FILE: FT06F001  
ALLOCATIONS: RECORD FORMAT = NA  
RECORD LENGTH = 80 BYTES  
BLOCK SIZE = NA  
FILE NAME: TERMINAL SCREEN  
FUNCTION: OUTPUT TO TERMINAL SCREEN

7. FILE: OUTPUT  
ALLOCATIONS: RECORD FORMAT = NA  
RECORD LENGTH = 80 BYTES  
BLOCK SIZE = NA  
FILE NAME: TERMINAL SCREEN  
FUNCTION: OUTPUT TO TERMINAL SCREEN FROM PASCAL

8. FILE: OUTFIL  
ALLOCATIONS: RECORD FORMAT = FB(FIXED BLOCK)  
RECORD LENGTH = 80 BYTES  
BLOCK SIZE = 6160 BYTES  
FILE NAME: USERID.PDDI.TERMINAL.DATA  
FUNCTION: CHARACTER OUTPUT TO FILE

9. FILE: FT10F001  
ALLOCATIONS: RECORD FORMAT = FB(FIXED BLOCK)  
RECORD LENGTH = 80 BYTES  
BLOCK SIZE = 6160 BYTES  
FILE NAME: USERID.PDDI.HEX.DATA  
FUNCTION: HEXADECIMAL ADB OUTPUT TO FILE

### 2.3.1 Overview

```
*****  
*                               *  
*               PIES           *  
*               - - - -        *  
* PRODUCT INFORMATION EXCHANGE SYSTEM *  
*****  
  
ENTER THE NAME OF THE "PID" FILE/RETRIEVE DATASET.  
CAD5.PID.COMRIB.DATA  
ENTER THE NAME OF THE "EXCHANGE FILE" DATASET.  
CAD5.EF.COMRIB.DATA
```

```
*****  
*                                     *  
*                               PIES   *  
*                   -----          *  
*    PRODUCT INFORMATION EXCHANGE SYSTEM *  
*           CUM CPU = 0.1 SECS         *  
*                                     *  
* ENTER ONE OF THE FOLLOWING OPTIONS:  *  
*                                     *  
* "TRAN"   FOR MODEL TRANSLATION TO/FROM PDSES *  
* "PID"    FOR PDDI INTERIM DATABASE ACCESS  *  
* "QUERY"   FOR MODEL QUERY                  *  
* "VERIFY"  FOR MODEL VERIFICATION           *  
* "DSPLAY"  TO SEND OUTPUT TO THE SCREEN.    *  
* "EXIT"    TO EXIT PROGRAM                 *  
*****
```



July 1989

### 2.3.2 PIES Main Menu

The PIES options allow the user to perform several operations as shown in the last example. The "TRAN" (Translate) option allows the user to translate the model (PDDI/PDES EF files). The "PID" (PDDI Interim Database) option allows the user to file/retrieve a Working Form (WF) model using sequential files. The "QUERY" option allows the user to query the model data. The "VERIFY" option allows the user to verify the integrity of a WF Model. The "DSPLAY/PRINT" option allows the user to direct QUERY or VERIFY output to the terminal screen or a file. These operations are explained in the following sections.

### 2.3.3 System Translator

#### 2.3.3.1 Overview

The System Translator is used to transmit the PDD between systems using a PDES structured file. The Pre-processor portion of the System Translator provides the interface from the MAS WF to the PDES EF file. The Post-processor portion of the System Translator provides the interface from the PDES EF file to the MAS WF data.

When translating a model, the System Translator interfaces with the Data Dictionary. The Data Dictionary is used to map the WF entities to the PDES file structure. If no Data Dictionary entry exists for an entity, the System Translator does not recognize it and it is not translated. Additionally, because the mapping from WF to PDES file structure (or vice-versa) is done at runtime, the user is able to translate any type of entity by simply providing a Data Dictionary entry for the entity. The System Translator, being data independent, has run with the PDDI or GMAP Data Dictionary. For this reason, it has been referred to as the PDDI System Translator or the GMAP System Translator.

#### 2.3.3.2 System Translator Menu

To initiate the System Translator, the user enters "TRAN" from the PIES Main Menu. The System Translator menu will be displayed, as shown in the following example. The user will be asked to identify which processor they wish - PRE or POST.

If the user enters PRE, a WF model must exist in memory. This can be accomplished by retrieving a PID model (refer to Section 2.3.4) before invoking the "TRAN" option. After selecting the Pre-processor, the user is prompted for username, organization name, model name, and any description to be placed in the header section of the EF file. A carriage return on the last line of the "comments" signifies that the user has entered all information and initiates translator processing. The Pre-processor produces

a PDES EF file in the Exchange File Dataset specified by the user. Refer to Section 2.2.

If the user enters POST, a WF model will be produced from the Exchange File Dataset specified by the user. The WF model may be saved for later use by using the PID function of PIES. For a detailed discussion of the PDDI System Translator, refer to the GMAP System Translator User's Manual (CI UM560240021U) and GMAP System Compents As-Built Product Specification (CI PS560240032U).

```
TRAN
PDES EXCHANGE FORMAT/WORKING FORMAT TRANSLATOR
VERSION: 2.2

DO YOU WISH TO RUN THE "PRE" PROCESSOR (WF -> EF)
OR "POST" PROCESSOR (EF -> WF)?

PRE
ENTER TRANSLATION TRANSACTION IDENTIFYING INFO!
ENTER YOUR NAME
JOHN DOE
ENTER YOUR ORGANIZATION
THE WIDGET CORPORATION
ENTER
CAD5.EF.COMRIB.DATA
ENTER FILE DESCRIPTION. HIT "RETURN" TO END.
THIS FILE IS AN EXAMPLE FROM MR. DOE. IT IS INTENDED TO
SHOW INDIVIDUALS HOW TO ACCESS THE PDDI TRANSLATOR FROM
THE PIES SOFTWARE.
```

#### 2.3.3.3 Exchange Format File Description

As previously mentioned, the EF File that is created and used in the System Translator conforms to the PDES file structure. An example of this file follows. For a detailed description of the defined structure of the PDES EF File, refer to STEP document 4.2.1 Version 11 dated June 1988.

HEADER/

DATE/19871120.071053;AUTHOR/PURSES;NAME/PDES;ORG/MDC;STATS/((:ADMINISTRATION,1  
:),(:APPROVAL,3:),(:CHARACTERISTIC,1:),(:DETAIL\_MODEL,1:),(:EFFECTIVITY,6:)  
,(:MATERIAL,1:),(:NOTE,15:),(:NEXT\_ASSEMBLY,1:),(:PLY\_DETAIL,3:),(:PLY,3:)  
,(:PLY\_TABLE,1:),(:LAMINATE,1:),(:COMP\_FLANGE,2:),(:COMP\_FLAT\_PAT,3:)  
,(:SEGMENT,98:),(:EDGES,341:),(:IMPL\_T\_HOLE,6:),(:THRU\_HOLE,6:),(:TEXT\_LINE,30  
:),(:COORDINATE,7:),(:VECTOR,559:),(:POINT\_VECTOR,24:),(:IPLANE,3:),(:POINT,14  
8:))

,(:CUBIC,148:),(:CURVE\_STRING,14:),(:LINE,61:),(:PARM\_BI\_CUBIC,56:)  
,(:RULED\_SURFACE,16:),(:VERTEX,142:),(:EDGE,219:),(:LOOP,77:),(:FACE,77:)  
,(:SHELL,1:),(:OBJECT,1:),(:SIZE,3:));

(\* \*)

(\* MCDONNELL DOUGLAS \*)

(\* 11/20/87 \*)

(\* 07:11:54 \*)

(\* PURSES \*)

(\* TM360JP \*)

(\* COMPOSITE RIB \*)

(\* \*)

ENDSEC;

DATA/

1=EFFECTIVITY/1,'AFA18A', '0001', '&UP', 'P';

2=EFFECTIVITY/1,'ATFA18A', '0001', '&UP', 'P';

3=EFFECTIVITY/1,'CF18A', '0001', '&UP', 'P';

4=EFFECTIVITY/1,'CF18B', '0001', '&UP', 'P';

5=EFFECTIVITY/1,'F/A18A', '0001', '&UP', 'P';

6=EFFECTIVITY/1,'TF/A18A', '0001', '&UP', 'P';

7=APPROVAL/'1/12/85', 'E.B. BIRCHFIELD', 'PDDI PROJECT MANAGER

,

'352';

8=APPROVAL/'1/12/85', 'W.D. MCGOWAN', 'PDDI PART SUPERVISOR

,

'352';

9=APPROVAL/'1/12/84', 'D.S. GORDON', 'PDDI COMP PART DESIGNER

,

'352';

10=POINT/0,(0,0,0),0,0,0,-1.5185969623085E+00,

8.5504399420080E+00,-1.8199991962829E-01;

.

.

2076=COMP\_FLANGE/0,(0,0,0),1,0,(#1838,#1902,  
#1840,#1904,#1846,#1906,#1908,#1910,#1912,#1914,#1916,#1918,#1920,#1922);

2077=SHELL/0,(0,0,0),1,0,(#1886,#1932,#1888,  
#1850,#1878,#1880,#1890,#1852,#1882,#1884,#1892,#1854,#1894,#1934,#1896,#1856,

#1898,#1936,#1900,#1858,#1954,#1956,#2072,#1952,#1950,#2051,#2068,#1966,#1842,

```
#1838,#1876,#1902,#1844,#1840,#1874,#1904,#1848,#1846,#1872,#1906,#1938,#1908,  
#1870,#1910,#1940,#1912,#1868,#1914,#1942,#1916,#1866,#1918,#1944,#1920,#1864,  
#1922,#1946,#1924,#1862,#1926,#1948,#1928,#1860,#1930,#1964,#1962,#2075,#1960,  
#1958);  
2078=OBJECT/0,(0,0,0),1,0,#2077,;  
2079=DETAIL_MODEL/2.9999997466803E-02,2.50000000000000E-01,'CMP','R',0,  
'74A180607-1001 ','','76301',  
'RIB ASSY - INBD CLOSURE, AFT, FLAP, WG, LDG',  
' ',' ',' ',#1158,(#2018),#2019,(#1159),, (#2020,#2021,#2022,  
#2023,#2024,#2025,#2026,#2027,#2028,#2029,#2030,#2031,#2032,#2033,#2034), (#204  
1,#2042,#2049), (#2070,#2069,#2056,#2061,#2066,#2055,#2060,#2065,#2054,#2059,#2  
064,#2073,#2076,#2044,#2046,#2048,#2036,#2038,#2040),#2078;  
ENDSEC;
```

#### 2.3.4 PDDI Interim Database

##### 2.3.4.1 Overview

PID provides the user the capability to quickly file and retrieve WF models to and from the disk. The PID software will file a WF model to a sequential file, or retrieve a file and recreate the WF model. PID can also be used to perform a direct transfer of PDD between similar computers (IBM-IBM, VAX-VAX, APOLLO-APOLLO, and so on). This software was originally developed to provide an interim database capability for the user until an interface to an organization's native database system was developed.

Limitations of models that can be filed/retrieved by PID are:

- o Maximum allowable instances of one entity kind per model = 10000
- o Maximum allowable unique entity kinds per model = 500
- o Maximum allowable Attribute Data Block (ADB) length per entity instance = 45K bytes
- o Maximum allowable model size/file size = computer region/sequential file allocations available.

These limitations were chosen to maintain software performance levels while providing transportability. It is unlikely that these limitations will be reached for most models.

July 1989

#### 2.3.4.2 PDDI Interim Database Functions

##### 2.3.4.2.1 Access PDDI Interim Database

To access PID, the user should initiate execution of the PIES software, identify/allocate the PID dataset, and enter the "PID" command when the PIES Main Menu is displayed. At this point, the user is requested to enter "FILE" to file the WF model to disk, or "RTRV" to retrieve the WF model into computer memory, as shown in the following example.

```
PID
ENTER A "FILE" TO FILE THE MODEL
      "RTRV" TO RETRIEVE THE MODEL
RTRV

*****
***   PID MODEL RETRIEVAL   ***
*****

MODEL SUCCESSFULLY RETRIEVED
```

##### 2.3.4.2.2 File the Working Form Model

If the FILE operation "FILE" is selected, the memory resident WF model will be stored in the previously allocated PID formatted sequential dataset. The Working Form Model present in computer memory will still be available.

##### 2.3.4.2.3 Retrieve the Working Form Model

If the RETRIEVE operation "RTRV" is selected, the model from the PID formatted sequential dataset will be retrieved and stored in its memory resident WF. If this operation is selected, any previous WF model in computer memory will be replaced by the new model.

##### 2.3.4.3 THE PDDI Interim Database Software as a Separate Subsystem

The PID software can be used as a separate subsystem from PIES and can be linked into a user's application. To use it as a subsystem, the user must link the PID module into the using application module and allocate the PID dataset.

The file allocation required by PID is FT08F001. A description of this file was previously presented in Section 2.2.

User application calls are: "FILRTV(0,IRC)" to retrieve a WF model from disk, "FILRTV(1,IRC)" to file a WF model to disk, and "FILRTV(2,IRC)" to merge a WF model with another WF model currently in memory. IRC is a return code and indicates whether or not the file/retrieve/merge was successful. A return code of "0" indicates the operation was successful, and a return code of "1" indicates the operation was not successful.

For further details on the PID software, refer to the GMAP Software Installation Guide.

### 2.3.5 Query

#### 2.3.5.1 Overview

The QUERY function allows users to display to the terminal screen or print to a file the data contained in any entity within the WF model. The Data Dictionary is used to extract information from the WF entities. If a Data Dictionary representation does not exist for a particular entity kind, a hexadecimal dump of the entity ADB will be produced.

The Query function is initiated from the PIES main menu by keying in the command "QUERY". A Query Menu and its functions are shown in the following example.

#### QUERY

```
*****  
***          QUERY          ***  
*****
```

```
ENTER "MODSPA" TO PRODUCE A MODEL SPACE LISTING!  
ENTER "ALL"    TO QUERY ALL DATA!  
ENTER "KIND"   TO QUERY DATA BY ENTITY KIND!  
ENTER "NETREF" TO QUERY INCLUSIVE ENTITY REFERENCES!  
ENTER "NETDAT" TO QUERY INCLUSIVE ENTITY DATA!  
ENTER "ID"     TO QUERY DATA BY ENTITY INSTANCE!  
ENTER "HEXON"  TO DISPLAY/PRINT ADB IN HEXADECIMAL FORM.  HEX IS "OFF"!
```

July 1989

An example of the character Display/Print of an entity is shown in the following example. It shows the ADB Data, Constituent List References, and User List References. Constituent List and User List References are listed as "entity name (identifier number)." The IDENT number location in the ADB is automatically determined from the Data Dictionary entry for that entity. If an IDENT field cannot be found in the Data Dictionary entry, a default value of the fourth word of the ADB will be used.

```
*****
*****
* ENTITY NAME : PLY_DETAIL *
* 1 ENTITIES EXIST *
*****
*****
* 1 PLY_DETAIL ENTITIES ARE BEING PRINTED *

```

END OF PROCESSING

#### 2.3.5.2 Enter "HEXON" to Print Application Data Block in Hexadecimal Form

At the bottom of the Query Menu, an option exists to display/print the entity data in character form or hexadecimal form. If a Data Dictionary entry for an entity does not exist, the hexadecimal form is selected. However, this option allows for hexadecimal form even if a Data Dictionary entry exists.

The following example illustrates the hexadecimal display/print of an entity. Constituent and User references are listed in character form and the ADB data is shown in hexadecimal form. The task of determining the individual pieces of ADB data is left as an exercise for the user.

```
*****
THE KIND IS 12001. THE ADB LENGTH IS 43.
A HEX DUMP OF THE ENTITY ADB FOLLOWS:
BEGIN ADB

                                2001 E
002E0002000000000000000000000000440000FFFF400C444444
00E1000B000000040000000010000100000010200101E5000000
END ADB

THE CONSTITUENTS OF THIS ENTITY ARE:
CONSTITUENT 1 REFERENCES COORDINATE(0022)
CONSTITUENT 2 REFERENCES VECTOR(0399)
CONSTITUENT 3 REFERENCES ARRAY_ENTITY(0122)
*****
```

#### 2.3.5.3 Enter "MODSPA" to Produce a Model Space Listing

The "MODSPA" option is used to display/print a count of the entities in the WF model and the computer memory space used. This option is accessed from the Query Menu by keying in the command "MODSPA". The following examples illustrate a display/print resulting from this choice. If a Data Dictionary entry exists for a specific entity kind, then its entity name, followed by a kind number, will be listed with the model space data.



July 1989

Otherwise, the kind number alone will be displayed in place of an entity name.

QUERY

\*\*\*\*\*  
\*\*\* QUERY \*\*\*  
\*\*\*\*\*

ENTER "MODSPA" TO PRODUCE A MODEL SPACE LISTING!  
ENTER "ALL" TO QUERY ALL DATA!  
ENTER "KIND" TO QUERY DATA BY ENTITY KIND!  
ENTER "NETREF" TO QUERY INCLUSIVE ENTITY REFERENCES!  
ENTER "NETDAT" TO QUERY INCLUSIVE ENTITY DATA!  
ENTER "ID" TO QUERY DATA BY ENTITY INSTANCE!  
ENTER "HEXON" TO DISPLAY/PRINT ADB IN HEXADECIMAL FORM. HEX IS "OFF"!  
MODSPA

\*\*\*\*\*  
\*\*\* MODEL SPACE \*\*\*  
\*\*\*\*\*

149 ARRAY_ENTITY(1100)	= 7764	98 SEGMENT(1201)	= 4022
341 EDGES(1202)	= 11453	6 IMPL_T_HOLE(1203)	= 292
16 PICK_TOKEN(1301)	= 644	30 TEXT_LINE(1302)	= 4984
1 NIL_ENTITY(1307)	= 572	638 1315(1315)	= 30260
23 COORDINATE(3001)	= 1228	559 VECTOR(3002)	= 33592
24 POINT_VECTOR(3005)	= 1012	3 IPLANE(3006)	= 118
148 POINT(4001)	= 10772	148 CUBIC(5004)	= 8588
14 CURVE_STRING(5006)	= 688	61 LINE(5008)	= 2862
56 PARM_BI_CUBIC(6002)	= 3700	16 PRS(6010)	= 5284
142 VERTEX(8001)	= 7152	219 EDGE(8002)	= 11442
77 LOOP(8003)	= 3238	77 FACE(8004)	= 4883
1 SHELL(8005)	= 46	1 OBJECT(8006)	= 50
3 SIZE(9003)	= 160	1 ADMINISTRATION(10001)	= 89
3 APPROVAL(10002)	= 349	1 CHARACTERISTIC(10003)	= 142
1 DETAIL_MODEL(10004)	= 187	6 EFFECTIVITY(10005)	= 388
1 MATERIAL(10006)	= 166	15 NOTE(10007)	= 664
1 NEXT_ASSEMBLY(10009)	= 69	3 PLY_DETAIL(12001)	= 241
3 PLY(12002)	= 187	1 PLY_TABLE(12003)	= 54
1 LAMINATE(12004)	= 58	2 COMP_FLANGE(12005)	= 104
3 COMP_FLAT_PAT(12009)	= 166	6 THRU_HOLE(13006)	= 304

NUMBER OF ENTITIES = 2898  
SPACE USED BY ENTITIES = 161474

July 1989

#### 2.3.5.4 Enter "ALL" to Query All Data

If the user keys in the command "ALL" from the Query Menu, a character and/or hexadecimal display/print of every entity in the WF model will be generated. Every entity listed in the MODSPA query will be displayed/printed. The user is warned that this listing may be voluminous.

##### QUERY

```
*****
***          QUERY          ***
*****
```

```
ENTER "MODSPA" TO PRODUCE A MODEL SPACE LISTING!
ENTER "ALL"    TO QUERY ALL DATA!
ENTER "KIND"   TO QUERY DATA BY ENTITY KIND!
ENTER "NETREF" TO QUERY INCLUSIVE ENTITY REFERENCES!
ENTER "NETDAT" TO QUERY INCLUSIVE ENTITY DATA!
ENTER "ID"     TO QUERY DATA BY ENTITY INSTANCE!
ENTER "HEXON"  TO DISPLAY/PRINT ADB IN HEXADECIMAL FORM.  HEX IS "OFF"!
ALL
```

#### 2.3.5.5 Enter "KIND" to Query Data by Entity Kind

Selecting the "KIND" option displays/prints information about all entities of a given entity kind (i.e., all points, lines, and so on). When the user keys in the command "KIND", the user will be asked for the names or kind numbers of the entity types to be queried, as shown in the following example. Upon completion, PIES will list all data for all entity types entered.

July 1989

QUERY

```
*****
***          QUERY          ***
*****
```

```
ENTER "MODSPA" TO PRODUCE A MODEL SPACE LISTING!
ENTER "ALL"    TO QUERY ALL DATA!
ENTER "KIND"   TO QUERY DATA BY ENTITY KIND!
ENTER "NETREF" TO QUERY INCLUSIVE ENTITY REFERENCES!
ENTER "NETDAT" TO QUERY INCLUSIVE ENTITY DATA!
ENTER "ID"     TO QUERY DATA BY ENTITY INSTANCE!
ENTER "HEXON"  TO DISPLAY/PRINT ADB IN HEXADECIMAL FORM.  HEX IS "OFF"!
```

KIND

```
ENTER ENTITY "NAME" OR "KIND NO."
```

PLY DETAIL

```
ENTER ENTITY "NAME" OR "KIND NO."
```

#### 2.3.5.6 Enter "NETREF" to Query Inclusive Entity References

If the user keys the command "NETREF" (NETwork REFerence) into the Query Menu, as illustrated in the first following example, they will obtain an inclusive entity reference listing. The NETREF option outputs a descending chain of the given entity and its constituents, as shown in the second following example, or an ascending chain of the given entity and its users, as shown in the third following example. The user is first prompted to enter either "USER" or "CNST", depending on which listing is desired. The next prompts are for the entity name or kind number and the entity ID. Each constituent or user is listed by entity name, ID, and kind number along with a number indicating the constituent level or user level from the initial entity. The first set of constituents or users would be at level 1. The constituents or users of those constituents or users would be at level 2, and so on. Each level is indented to help distinguish the levels.

QUERY

\*\*\*\*\*  
\*\*\* QUERY \*\*\*  
\*\*\*\*\*

ENTER "MODSPA" TO PRODUCE A MODEL SPACE LISTING!  
ENTER "ALL" TO QUERY ALL DATA!  
ENTER "KIND" TO QUERY DATA BY ENTITY KIND!  
ENTER "NETREF" TO QUERY INCLUSIVE ENTITY REFERENCES!  
ENTER "NETDAT" TO QUERY INCLUSIVE ENTITY DATA!  
ENTER "ID" TO QUERY DATA BY ENTITY INSTANCE!  
ENTER "HEXON" TO DISPLAY/PRINT ADB IN HEXADECIMAL FORM. HEX IS "OFF"!

NETREF

ENTER "USER" IF YOU WANT THE INCLUSIVE USERS OF AN ENTITY!  
ENTER "CNST" IF YOU WANT THE INCLUSIVE CONSTITUENTS OF AN ENTITY!  
ENTER A CR TO RETURN TO THE MAIN MENU!

CNST

ENTER ENTITY "NAME/KIND#(ID)" EXAMPLE : LINE(25)  
PLY DETAIL(1)

\*\*\*\*\*  
\* THE INCLUSIVE CONSTITUENTS ARE: \*  
\*\*\*\*\*

1 PLY\_DETAIL(0001) - 12001  
2 COORDINATE(0022) - 3001  
2 VECTOR(0399) - 3002  
2 ARRAY\_ENTITY(0122) - 1100  
3 FACE(0030) - 8004  
4 PARM\_BI\_CUBIC(0025) - 6002  
5 CUBIC(0049) - 5004  
6 POINT(0027) - 4001  
6 POINT(0041) - 4001  
6 VECTOR(0097) - 3002  
6 VECTOR(0098) - 3002  
5 CUBIC(0051) - 5004  
6 POINT(0026) - 4001  
6 POINT(0042) - 4001  
6 VECTOR(0101) - 3002  
6 VECTOR(0102) - 3002  
5 CUBIC(0032) - 5004  
6 POINT(0026) - 4001  
6 POINT(0027) - 4001  
6 VECTOR(0063) - 3002  
6 VECTOR(0064) - 3002  
.  
.  
.  
.  
2 MATERIAL(0001) - 10006  
3 NIL\_ENTITY(0001) - 1307  
3 NIL\_ENTITY(0001) - 1307

\*\*\*\*\*  
\* THE INCLUSIVE USERS ARE: \*  
\*\*\*\*\*

1 PLY\_DETAIL(0001) - 12001  
2 ARRAY\_ENTITY(0148) - 1100  
3 DETAIL\_MODEL(0001) - 10004  
4 1315(1140) - 1315  
2 ARRAY\_ENTITY(0138) - 1100  
3 PLY\_TABLE(0001) - 12003  
4 ARRAY\_ENTITY(0148) - 1100  
5 DETAIL\_MODEL(0001) - 10004  
6 1315(1140) - 1315  
4 ARRAY\_ENTITY(0139) - 1100  
5 LAMINATE(0001) - 12004  
6 ARRAY\_ENTITY(0148) - 1100  
7 DETAIL\_MODEL(0001) - 10004  
8 1315(1140) - 1315  
6 1315(1171) - 1315  
4 1315(1170) - 1315  
2 ARRAY\_ENTITY(0125) - 1100  
3 PLY(0001) - 12002  
4 ARRAY\_ENTITY(0148) - 1100  
5 DETAIL\_MODEL(0001) - 10004  
6 1315(1140) - 1315  
4 ARRAY\_ENTITY(0134) - 1100  
5 LAMINATE(0001) - 12004  
6 ARRAY\_ENTITY(0148) - 1100  
7 DETAIL\_MODEL(0001) - 10004  
8 1315(1140) - 1315  
6 1315(1171) - 1315  
4 1315(1167) - 1315  
2 1315(1164) - 1315

#### 2.3.5.7 Enter "NETDAT" to Query Inclusive Entity Data

The "NETDAT" option requests NETWORK DATA information of a given entity. The user is prompted ENTITY "NAME" OR "KIND NO." and ENTITY "ID". The Display/Print of the output will be a combination of Query inclusive constituent references of an entity, and character display/print of an entity. An example of querying the model by entity constituent reference data is shown below.

##### QUERY

```
*****  
***          QUERY          ***  
*****
```

```
ENTER "MODSPA" TO PRODUCE A MODEL SPACE LISTING!  
ENTER "ALL"    TO QUERY ALL DATA!  
ENTER "KIND"   TO QUERY DATA BY ENTITY KIND!  
ENTER "NETREF" TO QUERY INCLUSIVE ENTITY REFERENCES!  
ENTER "NETDAT" TO QUERY INCLUSIVE ENTITY DATA!  
ENTER "ID"     TO QUERY DATA BY ENTITY INSTANCE!  
ENTER "HEXON"  TO DISPLAY/PRINT ADB IN HEXADECIMAL FORM.  HEX IS "OFF"!
```

##### NETDAT

```
ENTER ENTITY "NAME/KIND#(ID)"  EXAMPLE :  LINE(25)  
PLY DETAIL(1)
```



### 2.3.5.8 Enter "ID" to Query Data by Entity Instance

Data query by ID displays/prints information for that given entity. Multiple entity names or kind numbers and IDs may be entered. Each entity in the constituent chain, as shown in the Query inclusive constituent references of an entity, is listed below.

#### QUERY

```
*****  
***          QUERY          ***  
*****
```

```
ENTER "MODSPA" TO PRODUCE A MODEL SPACE LISTING!  
ENTER "ALL"    TO QUERY ALL DATA!  
ENTER "KIND"   TO QUERY DATA BY ENTITY KIND!  
ENTER "NETREF" TO QUERY INCLUSIVE ENTITY REFERENCES!  
ENTER "NETDAT" TO QUERY INCLUSIVE ENTITY DATA!  
ENTER "ID"     TO QUERY DATA BY ENTITY INSTANCE!  
ENTER "HEXON"  TO DISPLAY/PRINT ADB IN HEXADECIMAL FORM.  HEX IS "OFF"!  
ID  
ENTER ENTITY "NAME/KIND#{ID}"  EXAMPLE :   LINE(25)  
MATERIAL(1)  
ENTER ENTITY "NAME/KIND#{ID}"  EXAMPLE :   LINE(25)  
MATERIAL(6)  
ENTER ENTITY "NAME/KIND#{ID}"  EXAMPLE :   LINE(25)
```

### 2.3.6. Verify

#### 2.3.6.1 Overview

The VERIFY function allows the user to verify the accuracy of a WF model.

The Data Dictionary is used to determine the integrity of each entity in the model. If no Data Dictionary entry exists for the entity, the entity will not be verified. The types of verification performed on each entity in the WF models are:

July 1989

- o Flag entities with no users
- o Flag entities with incorrect ADB lengths
- o Flag entities with incorrect constituent references
- o Flag entities with incorrect constituent list lengths.

This function will flag the entities that would not translate properly through the PDDI System Translator. An example of the output produced by the VERIFY function is illustrated on the following page.

VERIFY

\*\*\*\*\*  
\*\*\* VERIFICATION \*\*\*  
\*\*\*\*\*

\*\*\*\*\* 23 CIRCULAR\_ARC ENTITIES ARE BEING VERIFIED \*\*\*\*\*

1. THE DEFINED CHARACTERISTICS BEING VERIFIED ARE:
  - AN ADB LENGTH OF 30 BYTES,
  - THE ENTITY HAS AT LEAST ONE USER, AND
  - A CONSTITUENT LIST LENGTH OF 3, OF WHICH THE CNSTS ARE
    - CONSTITUENT # 1(P0) CAN REFERENCE :  
POINT\_CLASS - 4000
    - CONSTITUENT # 2(P1) CAN REFERENCE :  
POINT\_CLASS - 4000
    - CONSTITUENT # 3(PM) CAN REFERENCE :  
COORDINATE - 3001

2. THE ENTITIES NOT MATCHING THIS CRITERIA ARE:

\*\*\*\*\* 41 LINE ENTITIES ARE BEING VERIFIED \*\*\*\*\*

1. THE DEFINED CHARACTERISTICS BEING VERIFIED ARE:
  - AN ADB LENGTH OF 30 BYTES,
  - THE ENTITY HAS AT LEAST ONE USER, AND
  - A CONSTITUENT LIST LENGTH OF 2, OF WHICH THE CNSTS ARE
    - CONSTITUENT # 1(P0) CAN REFERENCE :  
POINT\_CLASS - 4000
    - CONSTITUENT # 2(P1) CAN REFERENCE :  
POINT\_CLASS - 4000
2. THE ENTITIES NOT MATCHING THIS CRITERIA ARE:
  - LINE(0002) CONSTITUENT # 1 HAS IMPROPER KIND FLANGE(0073)
  - LINE(0078) HAS INCORRECT ADB LENGTH OF 32
  - LINE(0201) HAS NO USERS

.  
.  
.  
.

### SECTION 3

#### BATCH PROCESSING

##### 3.1 IBM Batch Processing of PIES

To run the PIES software in batch mode on the IBM, Job Control Language (JCL) similar to that in the example shown below must be executed. Please note that the EXEC statement executes a different program than the interactive processing. Minor changes were made to the PIES program to allow batch processing. PIESB was created for this purpose. The functionality of this program is the same as that of the interactive system. Explanations of the DD statements were given in Section 2.2.

The batch PIES input commands exist as an input stream of the JCL file "INPUT". The responses will be the same as those of the interactive process. The user inserts PIES commands into this input stream in the order that they wish to process. As with the interactive system, PIESB directs output to the files "OUTFIL" or "FT10F001".

```
//USERIDT JOB ACCTNUM,'PIES BATCH',
//      LIM=(5.0,20.0,60,100),REGION=3000K,MSGCLASS=T,
//      NOTIFY=USERID,CLASS=A
//*
//*      RUN STEP
//*
//RUNTERM EXEC PGM=PIESB
//STEPLIB DD DISP=SHR,DSN=CAD2.GMAP.PIES.LOAD
//DDFILE DD DISP=SHR,DSN=CAD5.GMAP.V40.DDDEFN.DATA
//DDINX DD DISP=SHR,DSN=CAD5.GMAP.V40.DDINDX.DATA
//OUTFIL DD DSN=USERID.GMAP.PIES.DATA,DISP=SHR
//EFFILE DD DSN=CAD5.EF.COMRIB.DATA,DISP=SHR
//FT08F001 DD DSN=CAD5.PID.COMRIB.DATA,DISP=SHR
//FT10F001 DD DSN=USERID.GMAP.HEX.DATA,DISP=SHR
//FT06F001 DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//OUTPUT DD SYSOUT=*
//INPUT DD *
DSPLAY
PID
RTRV
QUERY
MODSPA
$$$
TRAN
PRE
```

JOHN DOE  
THE WIDGET CORPORATION  
CAD5.EF.COMRIB.DATA  
THIS FILE IS AN EXAMPLE FROM MR DOE. IT IS INTENDED TO  
SHOW INDIVIDUALS HOW TO ACCESS THE PDDI TRANSLATOR FROM  
THE PIES SOFTWARE, IN BATCH

EXIT  
/\*

### 3.2 Virtual Architecture Extended (VAX) Batch Processing of PIES

To run the PIES software in batch mode on the VAX, Digital Command Language (DCL), similar to the example that follows, must be written. For batch processing two command files are needed, as opposed to the one for interactive processing. The functionality of the batch program is exactly the same as the interactive process. There are also several assign statements. For an explanation of these, refer to Section 2.2.

The batch PIES input commands exist as an input stream of the DCL file "INPUT." The responses will be the same as those of the interactive process. The user simply inserts PIES commands into this input stream in the order desired to process them. As with the interactive PIES, the batch PIES can direct output to the specified files "OUTFIL" or "FT10F001." However, in contrast to the interactive PIES terminal screen output, the similar batch PIES output is directed to the batch execution file "PIES BATCH.LOG."

#### RUN\_PIES\_BATCH.COM

```
$ :  
$ : THIS COMMAND FILE IS USED TO INITIATE THE PIES SOFTWARE  
$ :  
$ : PROMPT USER FOR NECESSARY INFORMATION  
$ :  
$ SET DEF DUA2:[MADDERN.PIES.COMFIL]  
$ PIDFILE = "PIDFILE"  
$ EXCHFIL = "EXCHFIL"  
$ WRITE SYS$OUTPUT ""  
$ WRITE SYS$OUTPUT ""  
$ WRITE SYS$OUTPUT ""  
$ WRITE SYS$OUTPUT "ENTER THE NAME OF THE PID FILE/RETRIEVE FILE"  
$ INQUIRE PIDFILE
```

```
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT "ENTER THE NAME OF THE ""EXCHANGE FILE"" FILE"
$ INQUIRE EXCHFIL
$ SET TERM/UPPER
$ SUBMIT/KEEP DUA2:[MADDERN.PIES.COMFIL]ASSIGN_PIES_BATCH.COM -
  /LOG_FILE=DUA1:[MADDERN.PIES.COMFIL]PIES_BATCH.LOG -
  /PARAM=('PIDFILE','EXCHFIL')
$ SET TERM/LOWER
$ Convert/FDL=dual:[v5.source]REBLOCK/Pad=" " EFILE EFILE
$ EXIT
```

### ASSIGN\_PIES\_BATCH.COM

```
$ SET VERIFY
$ SET DEF DUA2:[MADERN.PIES.COMFIL]
$ ASSIGN 'P1' PIDFIL
$ ASSIGN 'P2' EFILE
$ ASSIGN DUA1:[GMAP.V40.DDFILS]PDDI_DDD.DAT      DDFILE
$ ASSIGN DUA1:[GMAP.V40.DDFILS]PDDI_DDI.DAT      DDINX
$ ASSIGN GMAP_HEX.DAT                            HEXDAT
$ ASSIGN GMAP_PIES.DAT                            UTFIL
$ RUN/NODEBUG DUA2:[MADDERN.PIES.COMFIL]PIES.EXE
PID
RTRV
DSPLAY
VERIFY
EXIT
```

## SECTION 4

### PIES OPERATOR'S MANUAL

#### 4.1 Introduction

This section describes the system operating commands and software installation procedures for the PIES software. It is intended for use by computer operators and programming personnel. It is assumed that personnel installing this software are familiar with the "native" system installation requirements and procedures.

The PIES software was designed to be transportable and has been operated on IBM 43XX, 308X and DEC VAX 11/780 computers. The PIES system document does not address local (native) system or computing environment documentation.

#### 4.2 Operations

##### 4.2.1 Introduction

The exchange medium for the PIES software is typically one magnetic tape for each computer system. Refer to Table 4-1 for the IBM system, and Table 4-2 for the VAX system.

##### 4.2.2 Files

The PIES system uses a collection of files to provide and hold data used by the system. These files have been assigned logical names. Therefore, the names must be edited to reflect native system requirements.

##### 4.2.3 Operator Interface

From a terminal on the native system, the operator executes a series of CLISTS and JCL on IBM, or Command files on VAX, to enact installation. The basic functions performed during system installation are "compile" and "link".

##### 4.2.4 Compile

The compile phase produces object code from source code. This object code is then used in the linkage editor.

##### 4.2.5 Link

The IBM Linkage Editor produces the load module from the object code

generated during the compile phase. The VAX linker produces the executable image from the object library generated during the compile phase.

#### 4.3 Installation Procedures

##### 4.3.1 Introduction

The VAX Command files and IBM CLIST and JCL files which are provided for installation are the same used in the development of PIES. These files do not include native system procedures. Vendor hardware and operating system manuals provide the necessary instructions for basic functions, (i.e., loading tapes, native system start-up, and device initialization). Other procedures needed to operate a native system should be obtained prior to PIES installation.

##### 4.3.2 Preliminary Steps

The following preliminary steps should be taken for installation procedures:

- o Ensure that appropriate personnel are familiar with the operation of the "native system"
- o Ensure that adequate "native system" resources are available
- o Obtain the current GMAP System Software, Exchange Format Files, and PIES
- o Mount tapes and read the GMAP tapes.

##### 4.3.3 Procedures - PIES Software

###### A. PIES IBM Software Installation

1. In Table 4-3, all the necessary PIES and PID source, include, and data datasets are listed. In Table 4-4, all the necessary PIES JCL and CLIST datasets needed to compile, link, and execute the software, are listed. These datasets may need to be changed to reflect your system libraries, naming conventions, and syntax differences between MVS and VM.

**Note:** Any name changes to datasets listed in Table 4-3 will need to be reflected in the related Table 4-4 datasets.

2. Create an object dataset for the PIES and PID datasets listed in Table 4-3.



i.e., source: CAD5.GMAP.V40.PIESSRC  
object: CAD4.GMAP.V40.PIESOBJ

3. Compile the PIES source routines into an object dataset. The JCL to compile the source routines are in the dataset CAD5.GMAP.V40.CNTL.

To compile the Product Information Exchange System routines, use the IBM JCL:

#### COMPIES

To compile PID routines use the IBM JCL:

#### COMPID

Refer to the GMAP Operator's Manual (CI OM 560240001U) for further details.

4. Link the PID object into a subsystem module.

To link the PID subsystem, use IBM JCL:

#### LINKPID

5. Link the subsystem modules into an executable load module.

To link the PIES system, use IBM JCL:

#### LINKPIES

Note: The PIES system uses the Model Access Software (MAS) and the GMAP System Translator. The JCL assumes that these two subsystems have been previously linked into load modules. The GMAP Operator's Manual contains the necessary JCL to complete this.

6. Run the executable load module. The CLIST to run the executable load module is in the dataset CAD5.GMAP.V40.CLIST.

To run the PIES executable load module, use clist:

#### RUNPIES

## B. PIES VAX Software Installation

1. In Table 4-3, all the necessary PIES and PID source, include, and data files are listed. In Table 4-5, all the necessary PIES command files needed to compile, link, and execute the software are listed.

Note: Any changes in the naming conventions of the directories listed in Table 4-3 will need to be reflected in the related Table 4-5 command files.

2. Create an object directory for the PIES and PID source directories listed in Table 4-3.

1.e., source: [GMAP.V40.PIESSRC]  
object: [GMAP.V40.PIESOLB]

3. Compile the PIES and PID source files into object libraries. All command procedures to compile the source files are in the directory [GMAP.V40.COMFIL].

To compile the Product Information Exchange System routines, use the command procedure:

### COMPILE\_PIES

To compile the PDDI Interim Database routines, use the command procedure:

### COMPILE\_PID

4. Link the PIES and PID object libraries into an executable image. The command procedure to link the object libraries is in the directory [GMAP.V40.COMFIL].

To link the Product Information Exchange System, use the command procedure:

### LINK\_PIES

Note: The PIES system uses the Model Access Software (MAS) and the GMAP System Translator. The DCL assumes that these two subsystems have been previously compiled into object libraries. The GMAP Operator's Manual contains the necessary DCL to complete this.

5. Run the executable image. The command procedure to run the executable image is in the directory [GMAP.V40.COMFIL].

To run PIES interactively, use the command procedure:

# RUN\_PIES

TABLE 4-1

## PIES SOFTWARE TAPE JCL IBM/MVS

```
//TM360JPT JOB ACCNTNO,/PIES',NOTIFY=TM360JP,
//      REGION=3000K,MSGCLASS=T,LIM=(30,90,60,60)
//*FORMAT PR,DDNAME,=DEST=N002
//*MAIN CLASS=A
//S1 EXEC PGM=IEHMOVE
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(TRK,(8))
//TAPE1 DD UNIT=TAPE62,DISP=(,PASS),LABEL=(,SL),DCB=DEN=3,
//      VOL=SER=(PIES)
//DD1 DD UNIT=3380,DISP=SHR,VOL=SER=LH02D0
//SYSIN DD *
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PIES,01),      X
  DSNAME=CAD5.GMAP.V40.CLIST
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PIES,02),      X
  DSNAME=CAD5.GMAP.V40.CNTL
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PIES,03),      X
  DSNAME=CAD5.GMAP.V40.DATA
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PIES,04),      X
  DSNAME=CAD5.GMAP.V40.PIDINC
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PIES,05),      X
  DSNAME=CAD5.GMAP.V40.PIDSRC
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PIES,06),      X
  DSNAME=CAD5.GMAP.V40.PIESINC
COPY TODD=TAPE1,FROM=3380=LH02D0,TO=TAPE62=(PIES,07),      X
  DSNAME=CAD5.GMAP.V40.PIESSRC
/*
//
```

TABLE 4-2

PIES SOFTWARE TAPE DCL  
VAX/VM

```
$ SET VERIFY
$!
$! THIS WILL CREATE A TAPE OF THE PIES SOFTWARE
$!
$ DELETE TAPE_LIST.LST;*
$ INIT/DENSITY=1600 MUA0: PIES
$ MOUNT/FOREIGN MUA0: PIES
$
$! COPY COMMAND PROCEDURES FOR COMPILING, LINKING, AND RUNNING
$
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -
  [GMAP.V40.COMFIL]COMPILE_PID.COM;,-
  [GMAP.V40.COMFIL]COMPILE_PIES.COM;,-
  [GMAP.V40.COMFIL]INC_ASSIGN_PID.COM;,-
  [GMAP.V40.COMFIL]INC_ASSIGN_PIES.COM;,-
  [GMAP.V40.COMFIL]RUN_PIES.COM;,-
  [GMAP.V40.COMFIL]PIES_LINK.COM -
  MUA0:COMFIL.BKP/SAVESET
$
$! COPY PID INCLUDE FILES
$
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -
  [GMAP.V40.PIDINC]*.* -
  MUA0:PIDINC.BKP/SAVESET
$
$! COPY PID SOURCE FILES
$
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -
  [GMAP.V40.PIDSRC]*.* -
  MUA0:PIDSRC.BKP/SAVESET
$
$! COPY PIES INCLUDE FILES
$
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -
  [GMAP.V40.PIESINC]*.* -
  MUA0:PIESINC.BKP/SAVESET
$
$! COPY PIES SOURCE FILES
$
$ BACKUP/LOG/VERIFY/LIST=TAPE_LIST.LST -
  [GMAP.V40.PIESSRC]*.* -
```

Table 4-2 (contd.)

```

MUA0:PIESSRC.BKP/SAVESET
$
$ DISMOUNT/NOUNLOAD MUA0:
$ COPY/CONCATENATE TAPE_LIST.LST;* PIES_V40_TAPE.LST
$ SET NOVERIFY
$ EXIT

```

TABLE 4-3

PIES SOFTWARE COMPONENTS

<u>File Name</u>	<u>Hardware</u>	<u>Description</u>
[GMAP.V40.PIDINC]	VAX	Contains PID PASCAL include files.
CAD5.GMAP.V40.PIDINC	IBM	
[GMAP.V40.PIDSRC]	VAX	Contains PID source files.
CAD5.GMAP.V40.PIDSRC	IBM	
[GMAP.V40.PIESINC]	VAX	Contains PIES PASCAL include files.
CAD5.GMAP.V40.PIESINC	IBM	
[GMAP.V40.PIESSRC]	VAX	Contains PIES source files.
CAD5.GMAP.V40.PIESSRC	IBM	

TABLE 4-4

PIES IBM JCL AND CLISTS

<u>File Name</u>	<u>Description</u>
CAD5.GMAP.V40.CNTL(COMPID)	IBM/MVS JCL to submit a batch compilation of PID source.
CAD5.GMAP.V40.CNTL(COMPIES)	IBM/MVS JCL to submit a batch compilation of PIES source.
CAD5.GMAP.V40.CNTL(LINKPID)	IBM/MVS JCL to submit a batch link of the PID load module.
CAD5.GMAP.V40.CNTL(LINKPIES)	IBM/MVS JCL to submit a batch link of the PIES load module.
CAD5.GMAP.V40.CLIST(RUNPIES)	IBM/MVS CLIST to execute the PIES.

TABLE 4-5

PIES VAX COMMAND FILES

<u>File Name</u>	<u>Description</u>
[GMAP.V40.COMFIL]COMPILE_PID.COM	VAX/VMS Command File to compile PID source.
[GMAP.V40.COMFIL]COMPILE_PIES.COM	VAX/VMS Command File to compile PIES source.
[GMAP.V40.COMFIL]PIES_LINK.COM	VAX/VMS Command File to link PIES.
[GMAP.V40.COMFIL]RUN_PIES.COM	VAX/VMS Command File to run an interactive session of PIES.
[GMAP.V40.COMFIL]INC_ASSIGN_PID.COM	VAX/VMS Command File to assign the PID PASCAL include files to logical names.
[GMAP.V40.COMFIL]INC_ASSIGN_PIES.COM	VAX/VMS Command File to assign the PIES PASCAL files to logical names.

TABLE 4-5 (contd.)

[GMAP.V40.COMFIL]ASSIGN\_DIR.COM

VAX/VMS Command File to assign  
the PID and PIES directory  
names to logical names.

## SECTION 5

### PERFORMANCE

Several benchmarks have been run with the PIES software to ensure that the speed with which the files can be translated (PDDI/PDES Translator) and filed/retrieved (PID) are not burdensome to the user. Statistics on two very different parts developed during the PDDI Air Force Program are shown in Figure 5-1 and Figure 5-2. These statistics are in agreement with similar tests performed on GMAP demonstration parts. They reflect enhancements made to increase the speed and efficiency of the final version (4.0) of the GMAP/PDDI System Translator.



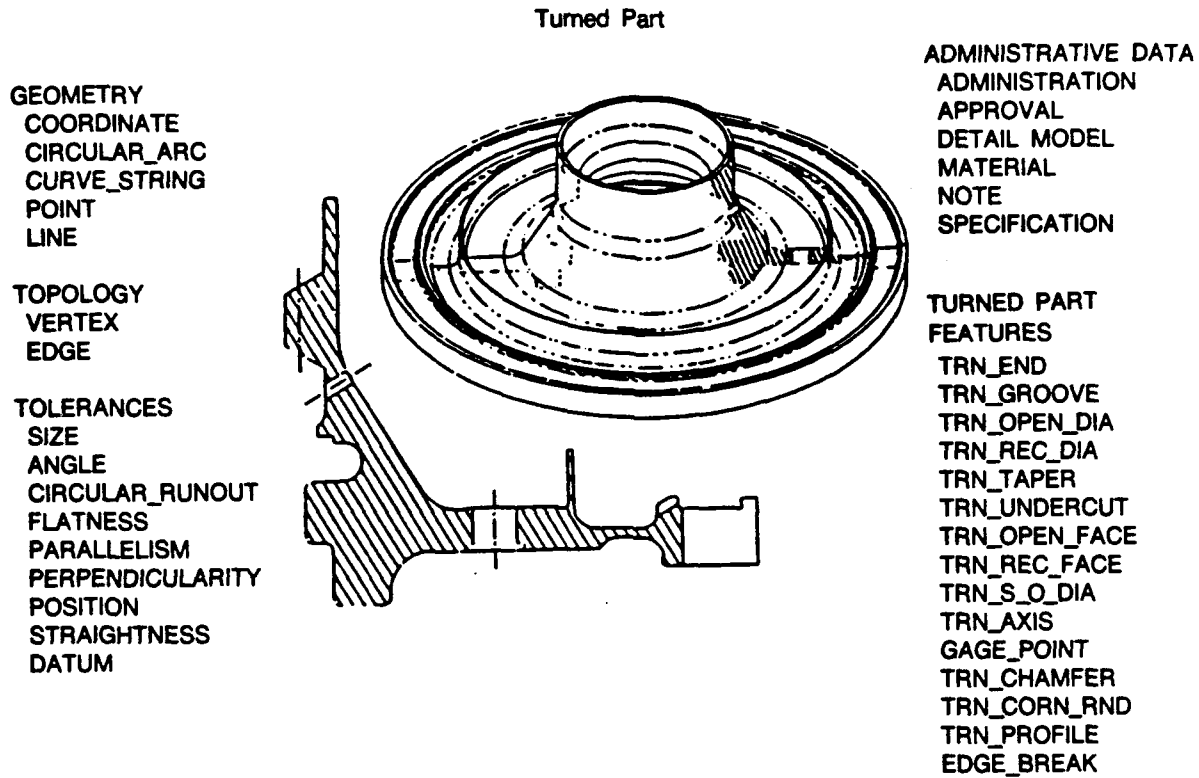


Figure 5-1. Translator and PID Performance on the Turned Part Model

NUMBER OF ENTITIES IN THE MODEL.....	687
WORKING FORM MODEL SIZE.....	38,056 BYTES
CPU TIME REQUIRED TO PRE-PROCESS USING PDDI TRANSLATOR.....	3.25 SECS
CPU TIME REQUIRED TO POST-PROCESS USING PDDI TRANSLATOR.....	3.00 SECS
CPU TIME REQUIRED TO FILE USING PID.....	1.51 SECS
CPU TIME REQUIRED TO RETRIEVE USING PID.....	1.80 SECS

**NOTE:** Statistics were generated on the PDDI turned part using a national advanced systems (NAS) computer (IBM 4381 look-alike)

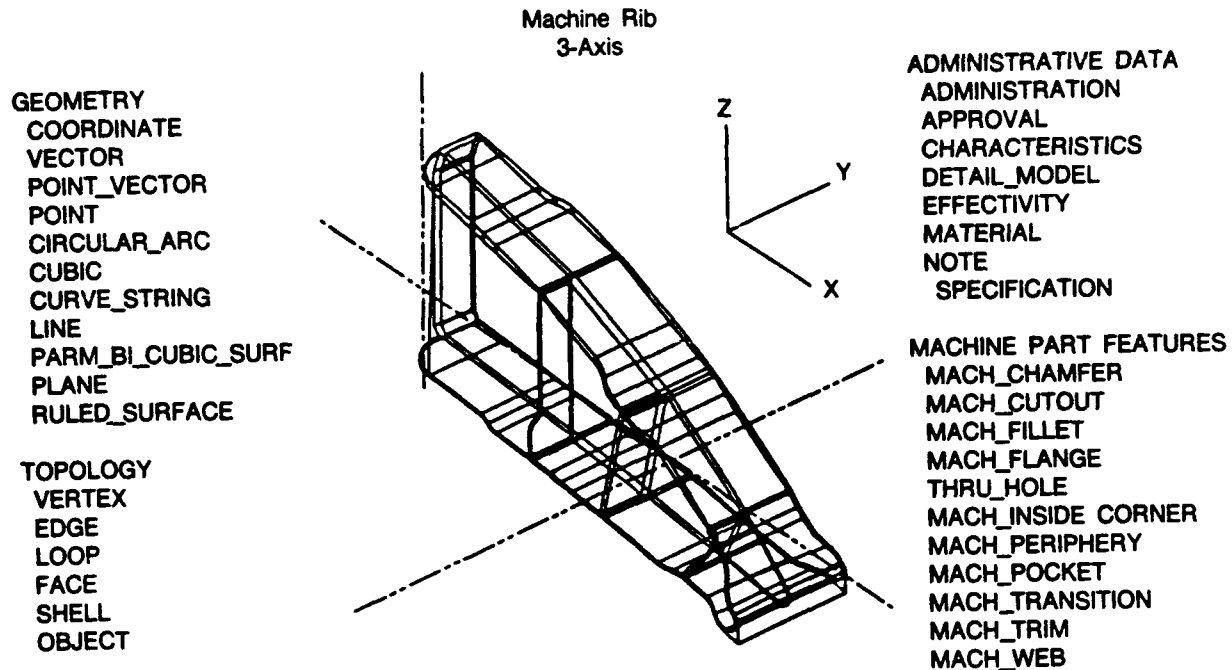


Figure 5-2. Translator and PID Performance on the Machine Rib Model

NUMBER OF ENTITIES IN THE MODEL.....	3,933
WORKING FORM MODEL SIZE.....	217,081 BYTES
CPU TIME REQUIRED TO PRE-PROCESS USING PDDI TRANSLATOR.....	11.56 SECS
CPU TIME REQUIRED TO POST-PROCESS USING PDDI TRANSLATOR.....	15.12 SECS
CPU TIME REQUIRED TO FILE USING PID.....	6.42 SECS
CPU TIME REQUIRED TO RETRIEVE USING PID.....	8.87 SECS

Note: Statistics were generated on the PDDI 3-axis machine rib using a national advanced systems (NAS) computer (IBM 4381 look-alike)

July 1989

APPENDIX A

## REFERENCES

A.1 Reference Documents

The following technical reports, specifications, standards, and other documents have been referred to or are relevant to this Model Access Software User's Manual.

A.1.1 Military:

Integrated Computer Aided Manufacturing (ICAM) Architecture, Vol. 4, Function Modeling Manual (IDEF0), USAF Report No. AFWAL-TR-81-4023, June 1981.

Integrated Computer Aided Manufacturing (ICAM) Architecture, Vol. 5, Information Modeling Manual (IDEF1), USAF Report No. AFWAL-TR-81-4023, June 1981.

Integrated Computer Aided Manufacturing (ICAM) Documentation Standards, IDS 150120000C, September 1983.

PDDI System Specification, Product Definition Data Interface (PDDI) Project 5601, Contract F33516-82-5036, July 1984.

PDDI System Specification-Draft Standard, Product Definition Data Interface (PDDI), Project 5601, Contract F33516-82-5036, July 1984.

Information Modeling Manual IDEF-Extended (IDEF1X) Integrated Information Support System (I<sup>2</sup>S), ICAM Project 6201, Contract F33615-80-C-5155, December 1980.

Interim Technical Report No. 1 (ITR560240001U)  
"Geometric Modeling Applications Interface Program" February 1986  
(Period 1 August 1985 - 31 October 1985).

Interim Technical Report No. 2 (ITR560240002U)  
"Geometric Modeling Applications Interface Program" May 1986  
(Period 1 November 1985 - 31 January 1986).

Geometric Modeling Applications Interface Program (GMAP) Scoping Document, CI SD560240001U, May 1986.

Interim Technical Report No. 3 (ITR560240003U)  
"Geometric Modeling Applications Interface Program" August 1986  
(Period 1 February 1986 - 30 April 1986).

Interim Technical Report No. 4 (ITR560240004U)  
"Geometric Modeling Applications Interface Program" November 1986  
(Period 1 May 1986 - 31 July 1986).

Geometric Modeling Applications Interface Program (GMAP) Needs Analysis  
Document, CI NAD560240001U, November 1986.

Interim Technical Report No. 5 (ITR560240005U)  
"Geometric Modeling Applications Interface Program" January 1987  
(Period 1 August 1986 - 31 October 1986).

Geometric Modeling Applications Interface Program (GMAP) System  
Requirements Document, CI SRD560240001U, February 1987.

Geometric Modeling Applications Interface Program (GMAP) State of the  
Art Document, CI SAD560240001U, March 1987.

Interim Technical Report No. 6 (ITR560240006U)  
"Geometric Modeling Applications Interface Program" May 1987  
(Period 1 November 1986 - 31 January 1987).

Geometric Modeling Applications Interface Program (GMAP) System  
Specification (Volumes I-IV), CI SS560240001U, July 1987

Interim Technical Report No. 7 (ITR560240007U)  
"Geometric Modeling Applications Interface Program," August 1987  
(Period 1 February 1987 - 30 April 1987).

Geometric Modeling Applications Interface Program (GMAP) System Design  
Specification, CI SDS560240001U, November 1987.

Geometric Modeling Applications Interface Program (GMAP) to Retirement  
for Cause Interface Development Specification, CI DS560240011U, November  
1987.

Geometric Modeling Applications Interface Program (GMAP) to Integrated  
Blade Inspection System Interface Development Specification, CI  
DS560240021U, November 1987.

Geometric Modeling Applications Interface Program (GMAP) to Retirement  
for Cause Interface As-designed Product Specification, CI PS560240011U,  
December 1987.

Geometric Modeling Applications Interface Program (GMAP) to Retirement  
for Cause Interface Unit Test Plan, CI UTP560240011U, December 1987.

Interim Technical Report No. 8 (ITR560240008U)  
"Geometric Modeling Applications Interface Program," December 1987  
(Period 1 May 1987 - 31 July 1987).

Interim Technical Report No. 9 (ITR560240009U)  
"Geometric Modeling Applications Interface Program," March 1988  
(Period 1 August 1987 - 31 October 1987).

Geometric Modeling Applications Interface Program (GMAP) System Test  
Plan, CI STP560240001U, March 1988.

Product Definition Data Interface (PDDI)/Geometric Modeling Applications  
Interface Program (GMAP) Deliverables Roadmap Document, March 1988.

Geometric Modeling Applications Interface Program (GMAP) to Integrated  
Blade Inspection System Interface Unit Test Plan, CI UTP560240021U,  
March 1988.

Geometric Modeling Applications Interface Program (GMAP) to Integrated  
Blade Inspection System Interface As-designed Product Specification, CI  
PS560240021U, March 1988.

Geometric Modeling Applications Interface Program (GMAP) System  
Component As-designed Product Specification, CI PS560240031U, March 1988.

Interim Technical Report No. 10 (ITR560240010U)  
"Geometric Modeling Applications Interface Program," August 1988  
(Period 1 November 1987 - 31 January 1988).

Interim Technical Report No. 11 (ITR560240011U)  
"Geometric Modeling Applications Interface Program," August 1988  
(Period 1 February 1988 - 30 April 1988).

Geometric Modeling Applications Interface Program (GMAP) to Retirement  
for Cause Interface User Operator Manual, CI U/OM560240011U, August 1988.

Interim Technical Report No. 12 (ITR560240012U)  
"Geometric Modeling Applications Interface Program" October 1988  
(Period 1 May 1988 - 31 July 1988).

Geometric Modeling Applications Interface Program (GMAP) to Retirement  
for Cause Interface Unit Test Report, CI UTR560240011U, November 1988.

Geometric Modeling Applications Interface Program (GMAP) to Integrated  
Blade Inspection System Interface Unit Test Report, CI UTR5602421U,  
November 1988.

Geometric Modeling Applications Interface Program (GMAP) System  
Translator User Manual, CI UM560240021U, November 1988.

Geometric Modeling Applications Interface Program (GMAP) to Retirement  
for Cause Interface As Built Product Specification, CI PS560240012U,  
February 1989.

Geometric Modeling Applications Interface Program (GMAP) to Integrated  
Blade Inspection System Interface As-built Product Specification, CI  
PS560240022U, February 1989.

Geometric Modeling Applications Interface Program (GMAP) System  
Components Operator's Manual, CI OM560240001U, February 1989.

Geometric Modeling Applications Interface Program (GMAP) to Integrated  
Blade Inspection System Interface User/Operator Manual, CI  
U/OM560240021U, February 1989.

Geometric Modeling Applications Interface Program (GMAP) Schema Manager  
User's Manual, CI UM560240011U, February 1989.

Interim Technical Report No. 13 (ITR560240013U)  
"Geometric Modeling Applications Interface Program" February 1989  
(Period 1 August 1988 - 31 October 1988).

Interim Technical Report No. 14 (ITR560240014U)  
"Geometric Modeling Applications Interface Program" July 1989  
(Period 1 November 1988 - 31 January 1989).

Geometric Modeling Applications Interface Program (GMAP) Model Access  
Software User Manual, CI UM560240031U, July 1989.

Geometric Modeling Applications Interface Program (GMAP) PDD Editor  
User/Operator Manual, CI U/OM560240031U, July 1989.

Demonstration Model Descriptions for Geometric Modeling Applications  
Interface Program (GMAP), CI TTD560240001U, July 1989.

Product Information Exchange System (PIES) User Manual for Geometric  
Modeling Applications Interface Program (GMAP), CI TTD560240002U, July  
1989.

#### A.1.2 Commercial

A Practical Guide to Splines, C. de Boor, Applied Mathematical Sciences,  
Vol. 27, Springer-Verlag.

Design of Database Structures, T. J. Teorey and J. P. Fry,  
Prentice-Hall, Inc., Englewood Cliffs, N.J.

Differential Geometry of Curves and Surfaces, M. P. de Carmo,  
Prentice-Hall, Inc., 1976.

IDEFlX Readers Reference, D. Appleton Company, December 1985.

Identification of Product Definition Data in a Manufacturing Enterprise  
-- A Case Study. R. Lessard, United Technologies Research Center and R.  
Disa, Pratt & Whitney, March 1986.

Use of Product Models in a CIM Environment, D. Koziol Emmerson and K.  
Perlotto, Pratt & Whitney, March 1987.

Technical Issues in Product Data Transfer, Richard Lopatka, Pratt &  
Whitney, September 1987.

Implementation of GMAP Technologies for Logistic Support Applications,  
Donald L. Deptowicz, Pratt & Whitney, January 1988.

Barriers to PDES Approval, Anthony Day, Sikorsky, and Richard Lopatka,  
Pratt & Whitney, April 1988.

PDD: Implementation Issues, Diane Emmerson and Priscilla Blasko, United  
Technologies Corporation, Proceedings of AUTOFACT '88, October 1988.

Geometric Modeling Applications Interface Program: A Prototype for  
Active File Exchange, Linda Phillips and Diane Emmerson, United  
Technologies Corporation, National Computer Graphics Association  
Conference, April 1989.

### A.1.3 Standards Organizations

ANSI Y14.5M, Dimensioning and Tolerancing.

"The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database  
Management Systems," Information Systems, Vol. 3, pp. 173-191, 1978.

The Second Draft Report of the Ad Hoc Committee on the Content and  
Methodology of the IGES Version 3 (The Second PDES Report), K. Brauner  
and D. Briggs, November 1984.

EXPRESS - A Language for Information Modeling, ISO, TC184/SC4/WG1,  
January 1986.

The STEP File Structure, ISO, TC184/SC4/WG1, January 1987.

Mapping from EXPRESS to Physical File Structure, ISO, TC184/SC4/WG1, January 1987.

## A.2 Terms and Acronyms

A glossary of terms frequently used in GMAP which may be included in this Model Access Software User's Manual is provided below. Some reference notes applicable to these definitions are presented after the glossary. A list of acronyms and abbreviations used in GMAP is also included in this section.

### A.2.1 Terms Used in GMAP

**Accept/Reject/Incomplete Notice** -- A display on the cell computer that indicates the final status of the engine disk.

Accept	=	Acceptable within tolerance specified by engine manufacturer
Reject	=	Rejected because of flaw(s) outside the range of acceptable tolerances
Incomplete	=	Part cannot be inspected

**Access Software** -- A set of routines for creating, managing and querying an incore Working Form model.

**Angular** -- An angular size tolerance is used to tolerance the size of an angular feature independent of its angular location along an arc.

**Application** -- A method of producing a specific result.

**Application Request** -- A request initiated by an application program, either through batch or interactive processing, which will interrogate the model through the PDDI Access Software to obtain or operate on specific information regarding the model and its components or elements.

**Application Requested Data** -- The data which fulfills the application's original request and which is in the proper format and readable by the application.

**Architecture** -- A design or orderly arrangement.

**ASCII** -- American Standard Code for Information Interchange.



**As-Is** -- The present condition.

**Attribute** -- A quality of characteristics element of any entity having a name and a value.

**B-Spline** -- A spline defined by a control polygon, B-spline basis functions, and an associated knot vector. A Bezier curve is a special case of a B-spline; a nurb is the most general case of a B-spline.

**Bezier Curve** -- A type of curve defined by a set of vertices called a control polygon and a set of basis functions. The basis functions are known as Bernstein polynomials. K vertices define a curve of order K-1.

**Binding** -- Establishing specific physical references to data structures for an application program; may be performed at compile time or at run time.

**Blend** -- A smooth, continuous transition from one surface to another.

**Boundary Representation** -- A topology imposed on 3-D geometric entities to yield a general solid model. That model describes an object by describing its boundary area.

**Body of Revolution (BOR) Representation** -- A topology in which an object is represented as the volume swept by a curve rotated about a line. This is a boundary representation in which the curve represents the surface area of the object.

**Bounded Geometry** -- Geometry that has limits defined by its mathematical domain or range.

**Calibration Block Parameters (Scale Factors)** -- Nondestructive test parameters used to adjust a specific cell. These parameters are obtained from the calibration blocks located at each cell.

**Circumferential** -- A circumferential tolerance specifies the tolerance zone within which the average diameter of a circular feature must lie. The average diameter is the actual circumference divided by pi (3.14159). A circumferential tolerance is a specific example of a peripheral or perimeter tolerance for a general curve.

**Class** -- A collection of entities that are alike in some manner.

**CLIST** -- IBM Command lists.

**Composite Curve** -- A group of curve segments that are  $C^0$  continuous.

**Compound Feature Representation** -- An enumerative feature representation in which at least one component is itself a feature. For example, a bolt hole circle might be represented as a list of individual hole features.

**Concentricity (Generic)** -- A concentricity tolerance specifies a cylindrical tolerance zone within which the axis of a feature must lie, where the axis of the zone coincides with the axis of the datum.

**Conceptual Schema** -- Formally specified global view that is processing independent, covering information requirements and formulation of independent information structures. A neutral view of data, usually represented in terms of entities and relations.

**Conic** -- A quadratic curve represented in the most general case by the equation:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0.$$

A conic may be a circle, line, ellipse, parabola, or a hyperbola depending on the coefficients, A, B, C, D, E, and F.

**Constraints (Generic)** -- An assertion to explicitly specify data meaning or semantics.<sup>1</sup> (Notes appear at the end of this section.)

**Context-Free Grammar** -- The syntax of the language gives a precise specification of the data without interpretation of it.

**Constituent** -- A specific instance of an entity that is used in the definition of some other entity.

**Data Dictionary** -- A catalog of all data elements in a design, giving their name, definition, format, source, and usage. May also include data types and value limits.

**Defining Airfoil Sections** -- A planar or conical section that depicts an airfoil profile. Defining airfoil sections are those that meet aerodynamic requirements. Other intermediate sections are added for Manufacturing purposes.

**Dimension** -- A part dimension is a quantifiable value expressing size, form; or location.

**Domain** -- The set of values permissible in a given context.

**Dynamic Allocation** -- The allocation (and de-allocation) of memory resources as required by the application. The opposite is static allocation where a fixed size segment of memory is available to the application.

July 1989

**Eddy Current Cell** -- Hardware used to perform an Eddy current inspection operation (surface flaws).

**Eddy Current Inspection** -- An inspection method used to detect internal potential flaws on a disk. It is based on the principle of sending electromagnetic signals to a target area on a part and detecting/interpreting reflection (Eddy current) from the target.

**Eddy Current Scan Plan** -- An interpreter code program controlling the Eddy current inspection of a particular geometry.

**Eddy Current/Ultrasonic Flaw Data Printout** -- A printout containing size and location information about specific flaw(s) (both critical and noncritical) associated with a particular part.

**Entity** -- A description of a person, place, or thing, about which information is kept.

**External Reference** -- A reference to some quantity of data that exists somewhere outside the scope of the immediate body of information.

**Feature** -- A part feature in the dimensioning and tolerancing context is a feature in the sense of ANSI Y14.5M, that is, a physical component portion of a part, such as a surface, hole, slot, and so on, that is used in a tolerancing situation. In the dimensioning and tolerancing context, a feature consists of individual or groups of basic shape elements used to define the physical shape of an item. This general dimensioning and tolerancing use of features is to be distinguished from Features. The word "features" alone implies dimensioning and tolerancing features. The term "form feature" is described below.

**Feature Pattern** -- A geometric pattern of occurrences of similar form features, for example, a circular pattern of scallops, a rectangular array of holes.

**Feature Representation (Generic)** -- A description of a form feature within the context of a geometric model.

**Feature Type** -- A name applied to a form feature that is suggestive of its shape and size, for example, hole, slot, web.

**Feature of Size (Generic)** -- A feature of size provides a geometric location capable of being referenced for use with datums and tolerances. A feature of size can be a GMAP feature, or other referenceable shape elements of a part model that are symmetric about a point, line, plane, axis, curve, and so on. When a feature of size is used in a relationship with a tolerance or datum, its feature of symmetry is the implied reference.

July 1989

**Flat Pattern Representation (Extrusion Representation)** -- A topology in which an object is represented as the volume swept by a planar polygon moving in a direction normal to its plane. The polygon may have internal polygon represent the surface area of the object.

**Flaw Characteristics** -- Location, length, width, depth, and nondestructive test parameters associated with a specific flaw.

**Flaw Data Packet** -- Packet containing nonevaluated flaw data. Note that the packet can contain zero flaws.

**Flaw Orientation** -- The direction of the major characteristic of the flaw with respect of the part coordinate system. (See the notes section at the end of this glossary.)

**Flaw Suspect Location** -- The coordinate location of a possible flaw detected during a survey mode inspection (six-axis position of ultrasonic cell, seven-axis position of Eddy current cell).

**Form Feature** -- A portion of a part's geometry that is useful to regard as an entity. In a boundary representation context, this is a subset of the part's surface area.

**Form Tolerance** -- Form tolerances are used to control the form of model features. A form tolerance specifies the amount that an actual features form may vary from nominal. Form tolerance include straightness tolerance, flatness tolerance, roundness/circularity tolerance, cylindricity tolerance, perpendicularity tolerance, parallelism tolerance, angularity tolerance, profile-of-a-line tolerance, profile-of-a-surface tolerance, circular-runout tolerance, true-direction tolerance, and mismatch tolerance.

**Functionality** -- (1) To show that the configuration item has fulfilled the specified requirements. (2) The receiving and sending systems can operate on the entity in the same manner with the same results within a pre-defined tolerance.

**Function Modeling** -- A description of a system in terms of a hierarchy of functions or activities, each level decomposing higher ones into greater detail. Functions are named by verbs; nouns related are declared as inputs, controls, outputs, and mechanisms.

**Geometric Element (Generic)** -- An instance of a geometric entity.

**Geometric Group** -- A group of geometric entities with a name.

**Geometric Model** -- A part description in terms of its underlying geometric elements. The model may be a wireframe, surface, or solid model.

**Geometric Pattern** -- A circular or rectangular pattern of geometric entities.

**Group Technology Code** -- An alphanumeric string identifying significant characteristics of a product, enabling group technology applications. Also known as Part Classification Code.

**Include File** -- PASCAL source code from another file or library included on the compilation of a PASCAL source file.

**Input Data** -- That information which the application needs to supply in order to interrogate or operate on the model. This data may assume only these forms prescribed by the PDDI Access Software specification.

**Inspection Cycle** -- A period for which nondestructive testing inspection requirements are defined.

**Inspection Cycle Zone** -- An entity that is composed of a unique combination of zone and inspection cycle.

**Inspection Module Operator** -- Refers to personnel operating RFC cell(s).

**Instrument Setting Adjustments** -- Nondestructive testing parameter adjustments automatically accomplished via pre- and post-calibration operations. These adjustments have to be accomplished within a predetermined tolerance.

**Internal Flaw** -- A subsurface anomaly.

**Internal Flaw Major Characteristic** -- A vector determined by an agreed upon method.

Example (1): The vector of greatest magnitude from the centroid to a boundary of the anomaly.

Example (2): A vector representing the major axis of the minimum ellipsoidal envelope encompassing the anomaly.

**Internal Flaw Tolerance** -- A unique combination of:

- (a) Internal flaw orientation range.
- (b) Serviceable internal flaw tolerance limits.
- (c) Repairable internal flaw tolerance limits.

July 1989

**Internal Flaw Tolerance Limit** -- A unique combination of:

- 7(a) Maximum diameter.
- (b) Maximum depth below surface.
- (c) Maximum thickness.

**Interpreted Request** -- Input data which has been appropriately modified to conform to the PDDI Access Software's internal data representation so that it may be further processed.

**Key Attribute** -- An attribute or combination of attributes having values that uniquely identify each entity instance.<sup>2</sup>

**Laminates Representation (Generic)** -- A topology in which an object is represented as layers of flat material of known thickness.

**Location Tolerance** -- Location tolerances specify the allowable variation in position of model features. Location tolerances include various forms of position tolerancing conventions. These are (true) position, concentricity, alignment, rectilinear location, and angular location.

**Logistics Support** -- The function of procuring, distributing, maintaining, replacing, and repairing material in support of a delivered product.

**Machine Coordinate Positions** -- The probe location with respect to machine coordinates.

**Machine Preset Data** -- Machine coordinate adjustments automatically accomplished via pre- and post-calibration operations. These adjustments have to be accomplished within predetermined tolerance.

**Metadata** -- Data about data. Defines the physical schema and record formats of the part data.

**Metamodel** -- A body of data that defines the characteristics of a data model or structure.

**Model** -- A collection of PDD that is transferable, displayable, accessible, and equivalent to a Part. The internal representation of the application data, as initiated and organized by the user. The model is also referred to as the Working Form.

**Model Network Definition** -- The set of rules and definitions which outline in detail the data structure whereby higher order entities may be composed of lower order entities, or constituents, and the lower order entities may be constituents of one or more higher order entities.

**Native System** -- The PDD and applications in a format that is unique to the database of a CAD system.

**Nondestructive Testing Parameters** -- Parameters used by the Eddy current and ultrasonic instruments (examples: amplitude, phase angle, gain, threshold, and so on).

**Nonconstructive Feature Representation (Explicit Feature Representation)** -- A feature representation that at least partially depends on a declaration that a face, or portion of a face, is "in" the feature.

**Nondestructive Testing Personnel** -- Personnel responsible for the generation of scan plans and derivation of applicable nondestructive testing instrument settings used in the scan plans.

**Nonshape Data** -- Produce definition data that cannot be represented by shape elements.

**Normal Forms** -- Conditions reflecting the degree of refinement and control over the relationships and entities in an information model.

**Numerical Control Program (Complete and Proposed)** -- Set of program instructions used to generate a probe path.

**Orientation Range** -- An envelope in which the major flaw characteristic must lie.

**Parse** -- The process of analyzing input strings (records) to identify fields and to verify that the data has a valid format.

**Part Blueprint** -- A blueprint provided by the engine manufacturer of a particular F100 engine disk.

**Physical Schema** -- Internal representation of data; the computer view that includes stored record format and physical ordering of stored records.

**PID File** -- A PID File is a copy of the Working Form filed to disk for temporary storage. The software that produces this capability (PID Code) is provided as an interim solution while a translator to the native database is in development.

**Polynomial Spline** -- A parametric spline of order 1, 2, or 3 defined by a set of  $N+1$  points. The spline is CX, CY, or CZ continuous and defined by coefficients such that:

$$x(i) = AX(i) + BX(i) * S + CX(i) * S^{**2} + DX(i) * S^{**3}$$

$$y(i) = AY(i) + BY(i) * S + CY(i) * S^{**2} + DY(i) * S^{**3}$$

$$z(i) = AZ(i) + BZ(i) * S + CZ(i) * S^{**2} + DZ(i) * S^{**3}$$

and a parameter space  $(T_0, T_1, \dots, T_n)$

where

$$T(i) < = u < = T(i+1)$$

$$S = u - T(i)$$

**Position Tolerance** -- A position tolerance (true position) specifies a tolerance zone within which the feature may vary in any direction.

**Post-processor** -- A phase of the translator where data is received from the Exchange Format and is converted to the Working Form.

**Pre-processor** -- A phase of the translator where data is taken from the Working Form and is converted to the Exchange Format.

**Primitive Constructive Feature Representation (Generic)** -- A constructive representation that is noncompound and that does not incorporate another feature. Such a representation must consist solely of overt construction information. Representation of a through hole by centerline and diameter is an example.

**Probe Blueprint** -- Blueprint of Eddy current probe supplied by the probe manufacturer.

**Product Definition Data** -- Those data "explicitly representing all required concepts, attributes, and relationships" normally communicated from Design throughout Manufacturing and Logistics Support. The data include both shape and nonshape information required to fully represent a component or assembly so that it can be analyzed, manufactured, inspected, and supported. They enable downstream applications, but do not include process instructions. These data are not always finalized at the design release; the manufacturing process can also add to the product model or generate derived manufacturing product models.

**Product Life Cycle** -- Includes design, analysis, manufacturing, inspection, and product and logistics support of a product.



**Product Model** -- A computer representation of a product.

**Product Support** -- The function that interprets customer requests for information and can provide the technical responses to the customer in the form of technical orders and instructions.

**Proprietary Part Flaw Data** -- Formatted dataset containing proprietary data defining size(s), maximums, and location(s) of critical flaw(s) (dimensional and locational tolerance).

**RAW.O File** -- A data file that uses a bi-cubic patch surface representation to define the surfaces of an airfoil.

**Ready Status** -- Go/No-Go decision.

**Relation** -- A logical association between entities.<sup>3</sup>

**Remount Decision** -- Decision to remount an engine disk.

**Replicate Feature Representation (Generic)** -- A description of a feature as being identical to another feature except for location. Mathematically, a replicate feature representation consists of the identification of another (necessarily constructive) feature plus a transformation.

**Robot Initialization Parameters** -- A set of nondestructive testing parameters used to initialize the robot on an Eddy current or ultrasonic cell.

**Rotational Sweep** -- A sweep in which the swept curve is rotated about a line (the "centerline" of the sweep).

**Ruled Surface (Generic)** -- A surface defined by a linear blend of two curves.

**Run System** -- The Translator subpackage which provides the communication interface between the user and the pre/Post-processors.

**Run-Time Subschema** -- A subset of the data dictionary information used at run-time by the access software to provide field data and check data.

**Scan Plan** -- Instructions that drive an inspection; these include inspection area geometry, ordered inspection path points, inspection probe selection, inspection path for each probe, mechanical commands that allow mechanical manipulator positioning, instrument setting, and all the variables needed for signal processing and flaw data acquisition during inspection.

**Scan Plan Specifications** -- Standards and procedures used in creating Eddy current and ultrasonic scan plans for the RFC system.

**Schema** -- Formal definition of information structure. See Conceptual Schema, Physical Schema, Run-time Schema.

**Shape** -- The physical geometry of a mechanical part, as distinguished from a computer description of that geometry. Where the difference is significant, the attitude is taken that shape is nominal or basic, with shape variations of tolerances grafted thereon

**Shape Data** -- Include the geometric, topological description of a product along with the associated dimensional tolerances and feature descriptions.

**Single Spatial Probe/Transducer Path** -- The starting and ending location of a single probe movement.

**Size Tolerance** -- Size tolerances specify the allowable variation in size-of-model features, independent of location. Size tolerances include circumferential, rectilinear size, and angular size.

**Solid Geometric Model (Shape Representation)** -- A computer description of shape. The description may be partial in the sense that not all aspects of part shape are indicated. For example, a body of revolution representation of a turned part may not describe the nonaxisymmetric<sup>4</sup> aspects of part geometry. A solid model must be complete and unambiguous in the sense that it describes a single volume in 3-D space.

**Solid Modeling** -- The creation of an unambiguous and complete representation of the size and shape of an object.

**Source Code** -- A computer program written in some language which is processed to produce machine code.

**Spline** -- A piecewise polynomial of order K, having continuity up to order K-1 at the segment joints.

**Squirter Blueprint** -- Blueprint of the squirter head that houses the ultrasonic transducer.

**Subface** -- A subface is a bounded portion of a face. It is defined by an underlying face, exactly one periphery closed curve and zero, one, or more internal closed curves that represent cutouts or holes in the region. The internal closed curve must not touch or intersect each other or the periphery closed curve and must be entirely contained within the periphery closed curve.

July 1989

**Surface Flaw** -- A surface anomaly.

**Surface Flaw Major Characteristic** -- A vector determined by an agreed upon method.

Example: A vector representing the major axis of the minimum elliptical envelope encompassing the anomaly in the plane of the surface.

**Surface Flaw Tolerance** -- A unique combination of:

- (a) Surface flaw orientation range.
- (b) Serviceable surface flaw tolerance limits.
- (c) Repairable surface flaw tolerance limits.

**Surface Flaw Tolerance Limit** -- A unique combination of:

- (a) Maximum length.
- (b) Maximum width.
- (c) Maximum depth.

**Sweep Surface** -- Surfaces formed by extruding or revolving a planar profile in space.

**Syntax** -- Grammar: A set of rules for forming meaningful phrases and sentences from words in a vocabulary.

**System Computer** -- VAX 11/780 and supporting peripheral hardware.

**System Constraints** -- Those hardware and software environmental constraints which will be imposed upon the PDDI Access Software that will limit its implementation and application. An example of such constraints might be the particular compiler used to compile the PDDI Access Software package.

**To-Be** -- The future condition possible, given a proposed capability.

**Tolerance (Generic)** -- The total amount by which something may vary. For mechanical product definition, tolerances can be shape tolerances, weight tolerances, finish tolerances and so on. In the context of GMAP, the term "tolerance" used alone implies shape tolerance. Other forms of tolerance (nonshape) are explicitly stated, for example, "finish tolerance." In a GMAP product model, tolerances occur without dimensions. As in the Product Definition Data Interface Program, model dimensions are implicit in the model geometry. Therefore, application of a tolerance implies a specific underlying dimension or geometric condition.

**Topology** -- A data structure that assembles geometric entities (points, curves, surfaces) into a solid geometric model.

**Transducer Blueprint** -- Blueprint of ultrasonic transducer supplied by the transducer manufacturer.

**Transfer Data** -- The data required to make an exchange of data between systems (i.e., delimiters, record counts, record length, entity counts, numeric precision).

**Translator** -- A software MECHANISM that is used for passing data between the Exchange Format and Working Form of the PDD.

**Ultrasonic Cell** -- Hardware used to perform ultrasonic inspection operation (internal flaws).

**Ultrasonic Inspection** -- An inspection method used to detect surface flaws on a disk. It uses ultrasonic waves through a stream of water to send and collect signals concerning an area targeted for inspection.

**Ultrasonic Scan Plan** -- Interpreter code program controlling the ultrasonic inspection of a particular geometry.

**Unbounded Geometry** -- Geometry represented parametrically, without limits, usually by coefficients to a defining equation.

**Unigraphics (UG)** -- A computer graphics system.

**User Function (UFUNC)** -- An interface to the UG database.

**Working Form** -- Product definition data information in machine-dependent data formats; an a memory resident network model.

**Zone** -- A physical area of the disk composed of zone components.

**Zone Component** -- A subface, face, or feature that constitutes a zone or element of a zone.

---

NOTES:

<sup>1</sup> T.J. Teorey and J.P. Fry, Design of Database Structures, 1st edition, Prentice-Hall, Inc., Englewood Cliffs, N.J., p. 463.

<sup>2</sup> Integrated Computer Aided Manufacturing (ICAM) Architecture, Vol. 5, Information Modeling Manual (IDEF1), USAF Report NO. AFWAL-TR-81-4023, June 1981, p. 212.

<sup>3</sup> Ibid., p. 214.

<sup>4</sup> Ibid., p. 211.

A.2.2 Acronyms Used In GMAP

ADB	---	Application Data Block (also referred to as Attribute Data Block).
AIMS	--	Automated IDEF Methodology System.
ANSI	--	American National Standards Institute.
ANT	--	Abstract of New Technology.
APT	--	Automatically Programmed Tools.
ATP	--	Automation Technology Products.
BOM	--	Bill of Materials.
BOR	--	Body of Revolution.
BPI	--	Bits per Inch.
BREP	--	Boundary Representation.
CAD	--	Computer Aided Design.
CAE	--	Computer Aided Engineering.
CAEDS	--	Computer Aided Engineering Design System.
CALS	--	Computer Aided Acquisition and Logistics Support.
CAM	--	Computer Aided Manufacturing.
CAM-I	--	Computer Aided Manufacturing--International.
CAPP	--	Computer Aided Process Planning.
CAS	--	Cooled Airfoil System.
CDM	--	Common Data Model.
CDR	--	Critical Design Review.
CDT	--	Component Design Technology.
CFSR	--	Contract Fund Status Report.
CI	--	Configuration Item.
CIM	--	Computer Integrated Manufacturing.
CLIST	--	IBM command list.
CM	--	Configuration Management.
CMM	--	Coordinate Measuring Machine.
C/SSR	--	Cost/Schedule Status Report.
CWBS	--	Contract Work Breakdown Structure.
DBMS	--	Data Base Management System.

July 1989

DCL -- DEC Command Language.  
DDL -- Data Definition Language.  
DEA -- Digital Equipment Automation.  
DEC -- Digital Equipment Corporation.  
DESØ -- (ICAM) Architecture of Design.  
DJR -- Design Job Request; Drafting Job Request.  
DoD -- Department of Defense.  
DS -- Design Specification.  
DSM -- Design Substantiation Memo.  
EBCDIC -- Extended Binary Coded Decimal Interchange Code (IBM character set).  
EC -- Eddy Current.  
ECO -- Engineering Change Order.  
EDM -- Electrical Discharge Machining.  
EF -- Exchange Format.  
EII -- Engineering Information Index.  
EMD -- Engineering Master Drawing.  
EPCS -- Engine Product Configuration Support.  
ESA -- Engineering Source Approval.  
ESP -- Experimental Solids Proposal.  
FEDD -- For Early Domestic Dissemination.  
FEM -- Finite-Element Modeling.  
FOF -- Factory of the Future.  
FOS -- Feature of Size.  
FPIM -- Fluorescent Penetrant Inspection Module.  
FSCM -- Federal Supply Code for Manufacturers.  
GE -- General Electric.  
GMAP -- Geometric Modeling Applications Interface Program.  
GSE -- Ground Support Equipment.  
HCF -- High-Cycle Fatigue.  
IBIS -- Integrated Blade Inspection System.  
IBM -- International Business Machines.

ICAM	--	Integrated Computer Aided Manufacturing.
ICOM	--	Input/Control/Output/Mechanism.
ICS	--	Information Computer System.
IDEF	--	ICAM Definition.
IDEF0	--	IDEF Function Modeling.
IDEF1	--	IDEF Information Modeling.
IDEF1X	--	IDEF Extended Information Modeling.
IDEF2	--	IDEF Dynamics Modeling.
IDSS	--	Integrated Decision Support System.
IEEE	--	Institute of Electrical and Electronics Engineers.
IEN	--	Internal Engineering Notice.
IFS	--	Interface Specification.
IGES	--	Initial Graphics Exchange Specification.
IISS	--	Integrated Information Support System.
ILC	--	Improved Life Core.
IMS	--	Information Management System.
IPGS	--	(IBIS) Inspection Plan Generation System.
IRB	--	Industry Review Board.
IRIM	--	Infrared Inspection Module.
ISO	--	International Standards Organization.
ITA	--	Intelligent Task Automation.
ITI	--	International TechneGroup Incorporated.
ITR	--	Interim Technical Report.
LCF	--	Low-Cycle Fatigue.
MAS	--	Model Access Software.
MCAIR	--	McDonnell Douglas Corporation/McDonnell Aircraft Company.
MFG0	--	(ICAM) Architecture of Manufacturing.
MRP	--	Materials Requirements Planning.
NAD	--	Needs Analysis Document.
NBS	--	National Bureau of Standards.
N/C	--	Numerical Control.
NDE	--	Nondestructive Evaluation.

NDML	-- Neutral Data Manipulation Language.
NDT	-- Nondestructive Test.
NTSB	-- National Transportation Safety Board.
NVI	-- Name/Value Interface.
OGP	-- Optical Gaging Products, Inc.
PA/QA	-- Product Assurance/Quality Assurance.
PD	-- Product Data.
PDD	-- Product Definition Data.
PDDI	-- Product Definition Data Interface Program.
PDES	-- Product Data Exchange Specification.
PDL	-- Program Design Language.
PED	-- Preliminary Engine Design.
PI	-- Principal Investigator.
PID	-- PDDI Interim Database.
PIES	-- Product Information Exchange System.
PMP/PMS	-- Program Management Plan/Project Master Schedule.
PROCAP	-- Process Capability.
PS	-- Product Specification.
RFC	-- Retirement for Cause.
RPM	-- Revolutions per Minute.
SA-ALC	-- San Antonio-Air Logistics Center.
SAD	-- State-of-the-Art Document.
SD	-- Scoping Document.
SDL	-- Source Data List.
SDS	-- System Design Specification.
SL	-- Salvage Layout.
SML	-- Source Material Log.
SOA	-- State-of-the-Art (Survey).
SOR	-- Surface of Revolution.
SPC	-- Statistical Process Control.
SPF	-- System Panel Facility.
SQA	-- Software Quality Assurance.



SQAP	-- Software Quality Assurance Plan.
SRD	-- System Requirements Document.
SRL	-- Systems Research Laboratories.
SS	-- System Specification.
STEP	-- Standard for the Exchange of Product Model Data.
STP	-- System Test Plan.
TCTO	-- Time Compliance Technical Order.
TD	-- Technical Data.
TDCCR	-- Turbine Design Cost Reduction.
TDR	-- Tool Design Request.
TechMod	-- Technology Modernization.
TO	-- Technical Order.
TOP	-- Technical and Office Protocol.
TSO	-- Time-Sharing Option (IBM term).
UFUNC	-- User Function.
UG	-- Unigraphics.
UGFM	-- Unigraphics File Manager.
USA	-- Unified System for Airfoils.
USAF	-- United States Air Force.
UTC	-- United Technologies Corporation.
UTP	-- Unit Test Plan.
UTR	-- Unit Test Report.
UTRC	-- United Technologies Research Center.
VAX	-- Virtual Architecture Extended.
VMS	-- Virtual Memory System.
WBS	-- Work Breakdown Structure.
WF	-- Working Form.
WPAFB	-- Wright-Patterson Air Force Base.
XIM	-- X-Ray Inspection Module.